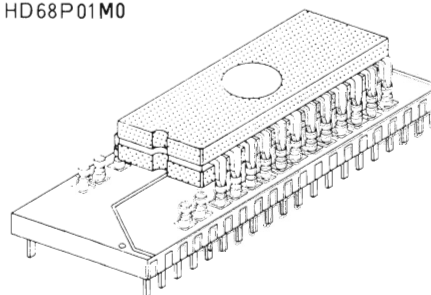
日立マイクロコンピュータシステム

HITACHI

HD68P01S0, HD68P01V05, HD68P01V07, HD68P01M0 MCU (Microcomputer Unit)

HD68P01は，HD6801ファミリとコンパチブルな命令オブジェクトコードを有し，同一チップ上に128バイトのRAM，シリアルコミュニケーションインタフェース(SCI)，パラレルI／O端子および3種類の機能を持つタイマを備え，かつパッケージ上にEPROM（2k，4k，8kバイト使用可）を搭載したシングルチップマイクロコンピュータユニットであり，HD6801ファミリとコンパチブルです。さらに，主要命令の実行時間の改良や，符号なし8ビット長データ同士の乗算（結果は16ビット長）を含む数種類の新操作命令が付加されており，システムスループットの向上が図られているほか，65kワードまでのアドレス空間をアクセスできます。また，最小の時間でファームウェアの変更を容易にし，かつ小規模量産システムにも適用できます。

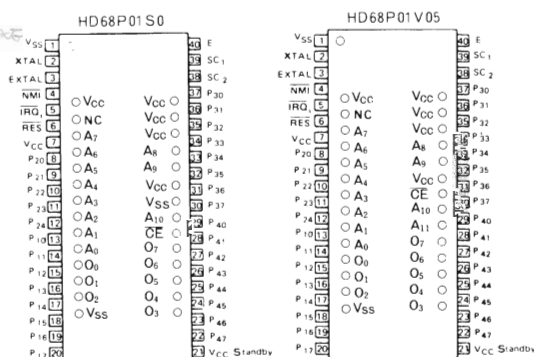
HD68P01S0，HD68P01V05，HD68P01V07
HD68P01M0

(DC-40P)

■特　長

- 命令セットはHMCS6800の拡張セット
- 8×8乗算機能

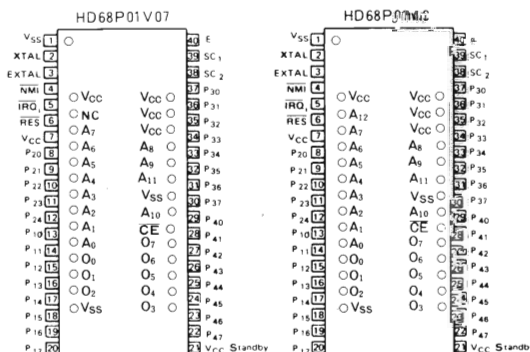
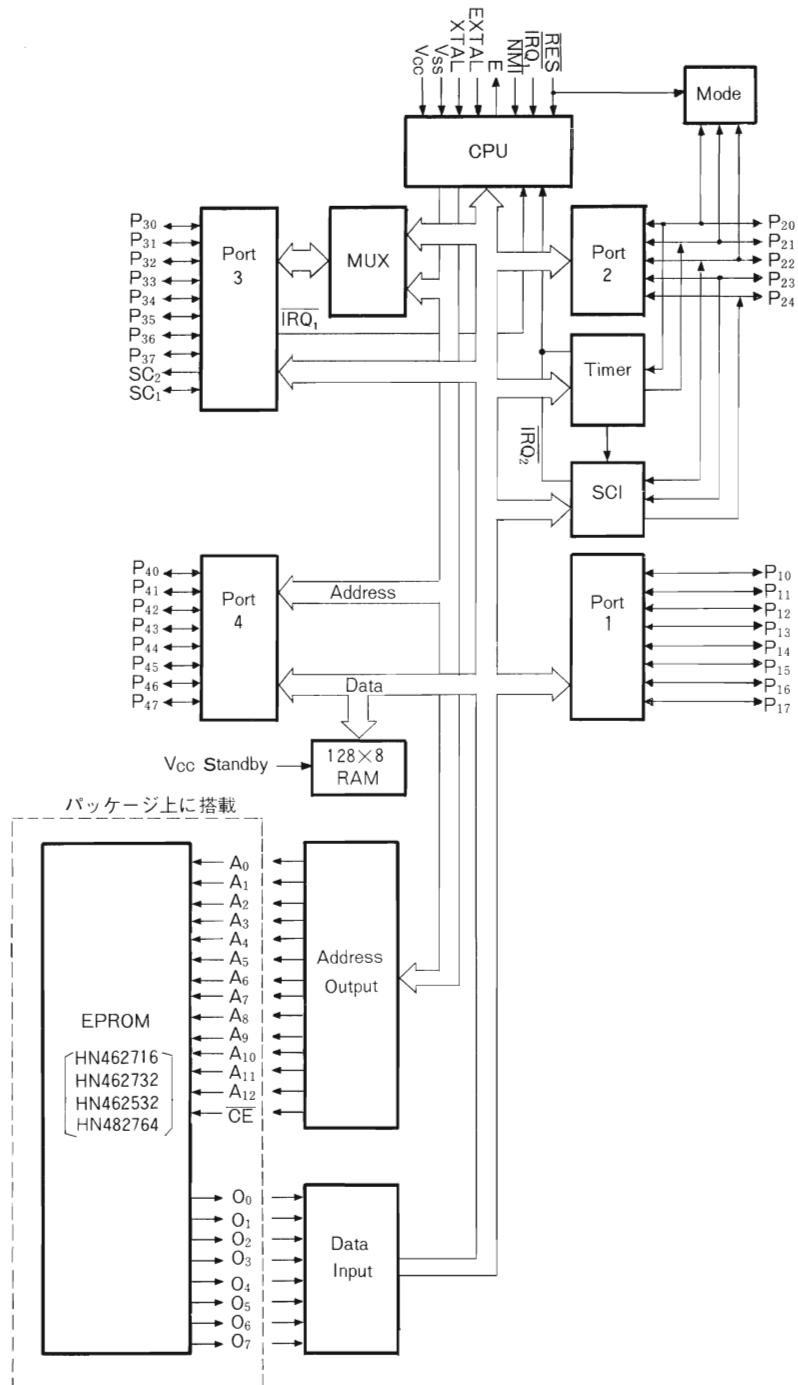
- シリアルコミュニケーションインタフェース(SCI)内蔵
- HD6800とコンパチブルなオブジェクトコード
- 16ビットタイマ内蔵
- すべての形式のEPROMを使用可
  - 2kバイト；HN462716相当
  - 4kバイト；HN462732，HN462532相当
  - 8kバイト；HN482764相当
- 128バイト内蔵RAM（先頭64バイトは電源オフ時内容保持可）
- 29本のパラレルI/O端子および2本のデータストローブ端子
- 発振回路を有し，4分周信号を内部クロックとして使用
- TTLコンパチブル入出力端子
- 強力な割込み機能

■ピン配置図（上面図）

（注）本MCUに搭載するEPROMは別売となっています。

■製品形名

形　　名	バスタイミング	適用EPROM
HD68P01S0	1MHz	HN462716相当
HD68P01V05	1MHz	HN462532相当
HD68P01V07	1MHz	HN462732相当
HD68P01M0	1MHz	HN482764相当

■ブロックダイアグラム

Vcc
Vss
XTAL
EXTAL
E
NMI
IRQ1
RES
Mode
CPU
Port 3
MUX
IRQ1
Port 2
Timer
IRQ2
SCI
P30
P31
P32
P33
P34
P35
P36
P37
SC2
SC1
P20
P21
P22
P23
P24
Port 4
Address
Data
Port 1
P40
P41
P42
P43
P44
P45
P46
P47
P10
P11
P12
P13
P14
P15
P16
P17
Vcc Standby
128×8 RAM
パッケージ上に搭載
EPROM
HN462716
HN462732
HN462532
HN482764
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
CE
Address Output
O0
O1
O2
O3
O4
O5
O6
O7
Data Input

■絶対最大定格

項　目	記　号	規　格　値	単　位
電源電圧	V_{CC}	−0.3〜+7.0	V
入力電圧	Vin	−0.3〜+7.0	V
動作温度	Topr	0〜+70	°C
保存温度	Tstg	−55〜+150	°C

（注）　本製品は高静電気電圧や高電界に対し，入力端子に保護回路を備えていますが，これらの高入力インピーダンス保護回路に最大定格以上の過電圧を印加しないように注意してください。また，パッケージ上面のピンソケットについても同様に注意してください。正常な動作を保証するためには，Vin，VoutをV_{SS}≦(V_{in}またはV_{out})≦V_{CC}にすることを推奨します。

■電気的特性

●DC特性（特記なき場合は V_{CC}=5.0V±5%，V_{SS}=0V，Ta=0〜+70°C）

項目		記号	測定条件	min	typ	max	単位
入力“High”レベル電圧	$\overline{RES}$	V_{IH}		4.0	−	V_{CC}	V
	その他の入力端子*			2.0	−	V_{CC}	
入力“Low”レベル電圧	全入力端子*	V_{IL}		−0.3	−	0.8	V
入力負荷電流	P_{40}〜P_{47}	$\|I_{in}\|$	Vin=0〜2.4V	−	−	0.5	mA
	SC_1			−	−	0.8	
	EXTAL		Vin=0〜V_{CC}	−	−	1.2	
入力リーク電流	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$	$\|I_{in}\|$	Vin=0〜5.25V	−	−	2.5	μA
スリーステート（オフ状態）リーク電流	P_{10}〜P_{17}, P_{30}〜P_{37}	$\|I_{TSI}\|$	Vin=0.5〜2.4V	−	−	10	μA
	P_{20}〜P_{24}			−	−	100	
出力“High”レベル電圧	P_{30}〜P_{37}	V_{OH}	I_{LOAD}=−205μA	2.4	−	−	V
	P_{40}〜P_{47},E,SC_1,SC_2		I_{LOAD}=−145μA	2.4	−	−	
	その他の出力端子		I_{LOAD}=−100μA	2.4	−	−	
出力“Low”レベル電圧	全出力端子	V_{OL}	I_{LOAD}=1.6mA	−	−	0.5	V
ダーリントン駆動電流	P_{10}〜P_{17}	$-I_{OH}$	V_{out}=1.5V	1.0	−	10.0	mA
消費電力		P_D		−	−	1200	mW
入力容量	P_{30}〜P_{37},P_{40}〜P_{47},SC_1	Cin	Vin=0V, Ta=25°C, f=1.0MHz	−	−	12.5	pF
	その他の入力端子			−	−	10.0	
スタンバイ電圧	非動作時	V_{SBB}		4.0	−	5.25	V
	動作時	V_{SB}		4.75	−	5.25	
スタンバイ電流	非動作時	I_{SBB}	V_{SBB}=4.0V	−	−	8.0	mA

＊モードプログラミングレベルは除く。

●AC特性

バスタイミング（特記なき場合はV_{CC}=5.0V±5%，V_{SS}=0V，Ta=0～+70°C）

項目	記号	測定条件	min	typ	max	単位
サイクル時間	t_{cyc}	図1，図2	1	—	10	μs
アドレスストローブパルス幅（“High” レベル）	PW_{ASH}		200	—	—	ns
アドレスストローブ立ち上がり時間	t_{ASr}		5	—	50	ns
アドレスストローブ立ち下がり時間	t_{ASf}		5	—	50	ns
アドレスストローブ遅延時間	t_{ASD}		60	—	—	ns
イネーブル立ち上がり時間	t_{Er}		5	—	50	ns
イネーブル立ち下がり時間	t_{Ef}		5	—	50	ns
イネーブルパルス幅（“High”レベル）	PW_{EH}		450	—	—	ns
イネーブルパルス幅（“Low”レベル）	PW_{EL}		450	—	—	ns
アドレスストローブからイネーブルまでの遅延時間	t_{ASED}		60	—	—	ns
アドレス遅延時間	t_{AD}		—	—	260	ns
ラッチのためのアドレス遅延時間	t_{ADL}		—	—	270	ns
データセットアップ時間（ライト時）	t_{DSW}		225	—	—	ns
データセットアップ時間（リード時）	t_{DSR}		80	—	—	ns
データホールド時間（リード時）	t_{HR}		10	—	—	ns
データホールド時間（ライト時）	t_{HW}		20	—	—	ns
ラッチのためのアドレスセットアップ時間	t_{ASL}		60	—	—	ns
ラッチのためのアドレスホールド時間	t_{AHL}		20	—	—	ns
アドレスホールド時間	t_{AH}		20	—	—	ns
ペリフェラルリードアクセス時間（非多重バス）	(t_{ACCN})		—	—	(610)	ns
ペリフェラルリードアクセス時間（多重バス）	(t_{ACCM})		—	—	(600)	ns
発振安定時間（パワーオン時リセット時間）	t_{RC}	図11	100	—	—	ms
プロセッサコントロールセットアップ時間	t_{PCS}	図10，図11	200	—	—	ns

ペリフェラルポートタイミング（特記なき場合はV_{CC}=5.0V±5%，V_{SS}=0V，Ta=0～+70°C）

項目		記号	測定条件	min	typ	max	単位
ペリフェラルデータセットアップ時間	ポート1,2,3,4	t_{PDSU}	図3	200	—	—	ns
ペリフェラルデータホールド時間	ポート1,2,3,4	t_{PDH}	図3	200	—	—	ns
遅延時間（イネーブル立ち上がりから$\overline{OS3}$の立ち下がり）		t_{OSD1}	図5	—	—	350	ns
遅延時間（イネーブルの立ち上がりから$\overline{OS3}$の立ち上がり）		t_{OSD2}	図5	—	—	350	ns
遅延時間（イネーブルの立ち下がりからペリフェラルデータ出力）	ポート1,2*,3,4	t_{PWD}	図4	—	—	400	ns
遅延時間（イネーブルの立ち下がりからペリフェラルデータ；CMOS）	ポート2**，4	t_{CMOS}	図4	—	—	2.0	μs
入力ストローブパルス幅		t_{PWIS}	図6	200	—	—	ns
入力データホールド時間	ポート3	t_{IH}	図6	50	—	—	ns
入力データセットアップ時間	ポート3	t_{IS}	図6	20	—	—	ns

* P_{21}は除きます。P_{21}は出力ポートとして使用できません。

** ポート2には，10kΩのプルアップ抵抗が必要です。

タイマ，SCIタイミング（特記なき場合はV_{CC}=5.0V±5%，V_{SS}=0V，Ta=0～+70°C）

項目	記号	測定条件	min	typ	max	単位
タイマ入力パルス幅	t_{PWT}		2tcyc+200	—	—	ns
遅延時間（イネーブルの立ち上がりからタイマ出力）	t_{TOD}	図7	—	—	600	ns
SCI入力クロックサイクル	t_{scyc}		1	—	—	tcyc
SCI入力クロックパルス幅	t_{PWSCK}		0.4	—	0.6	tscyc

モードプログラミング（特記なき場合はV_{CC}=5.0V±5%，V_{SS}=0V，Ta=0～+70°C）

項目	記号	測定条件	min	typ	max	単位
モードプログラミング入力“Low”レベル電圧	V_{MPL}	図8	—	—	1.7	V
モードプログラミング入力“High”レベル電圧	V_{MPH}		4.0	—	—	V
$\overline{RES}$パルス幅（“Low”レベル）	PW_{RSTL}		3.0	—	—	tcyc
モードプログラミングセットアップ時間	t_{MPS}		2.0	—	—	tcyc
モードプログラミングホールド時間	t_{MPH}		150	—	—	ns

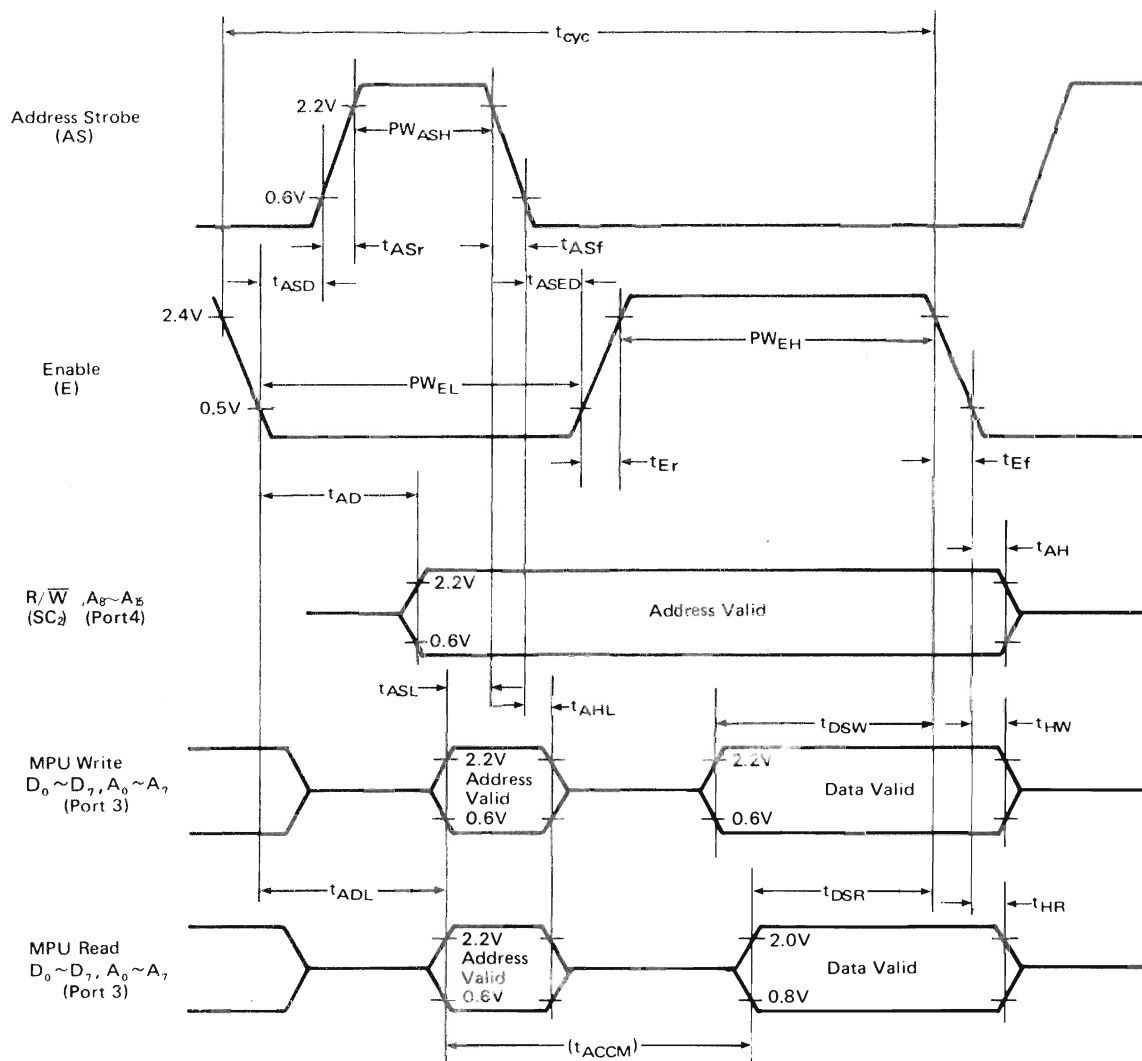



図1　エクスパンデッドマルチプレックスの場合のバスタイミング

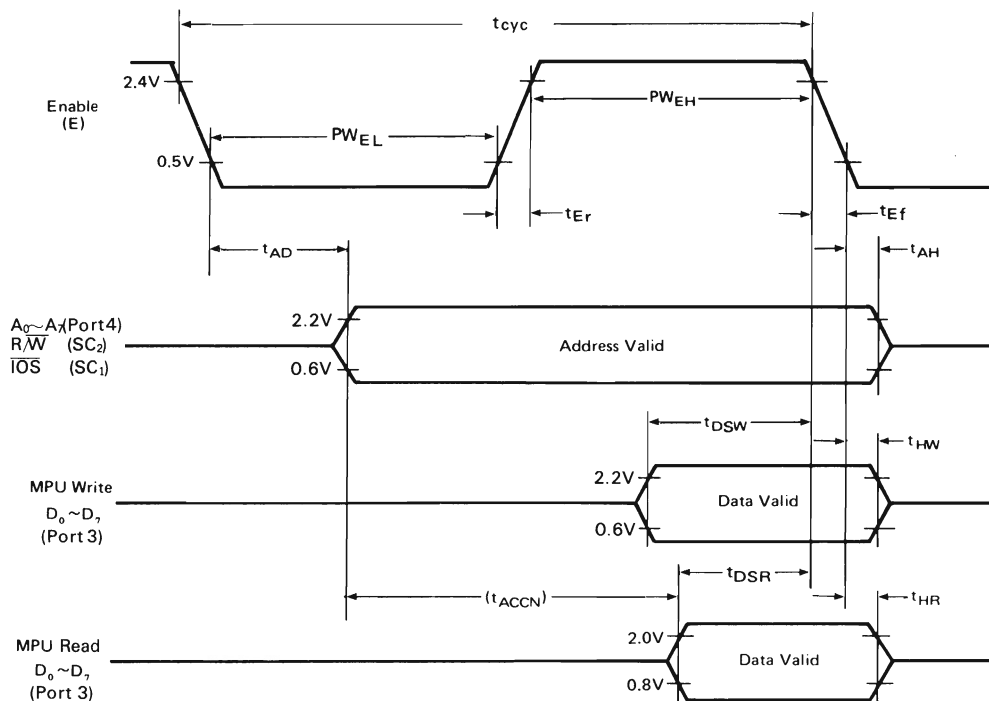



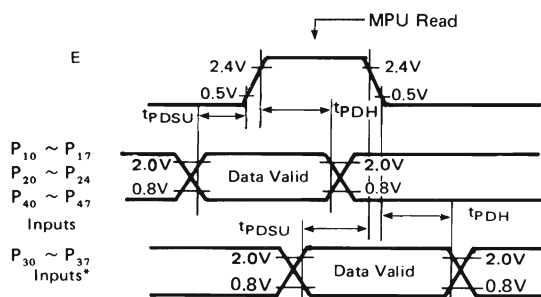
図2　エクスパンデッドノンマルチプレックスの場合のバスタイミング

* ポート3はラッチしない場合

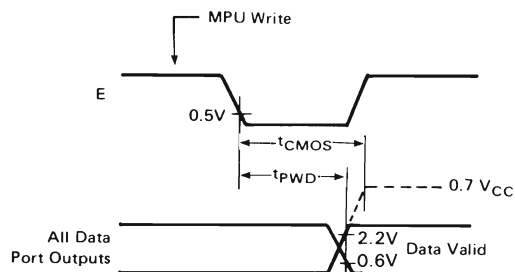
図3　データセットアップ時間とホールド時間（MPUリード）

(注)
1) ポート2のtCMOSは10kΩのプルアップ抵抗が必要。
2) P_{21}は除く
3) ポート4のtCMOSはプルアップ抵抗不要

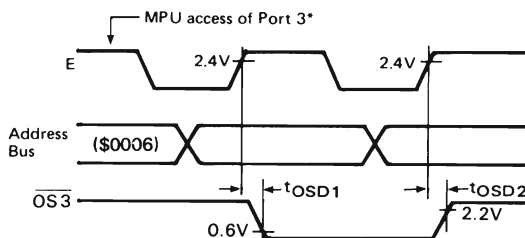
図4　データセットアップ時間とホールド時間（MPUライト）

* ポート3のアクセスがアウトプットストローブ選択ビットと一致したとき（OSS=0　リード時，OSS=1 ライト時）

図5　ポート3のアウトプットストローブタイミング（シングルチップモード）

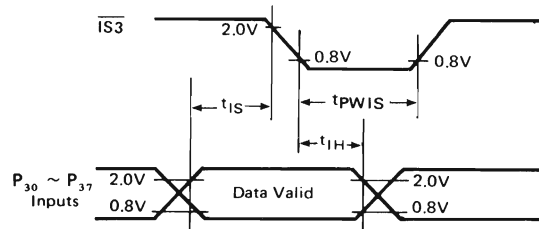



図6　ポート3のラッチタイミング（シングルチップモード）

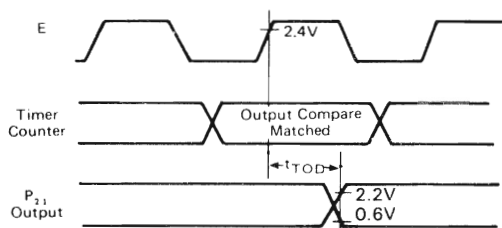



図7　タイマ出力タイミング

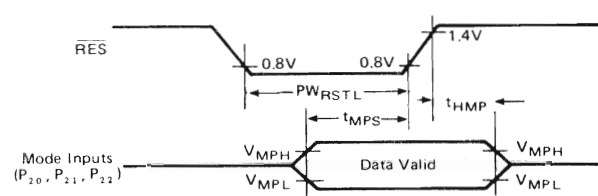



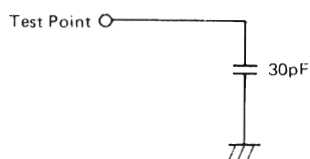
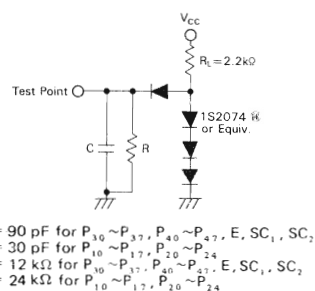
図8　モードプログラミングタイミング

(a) CMOS負荷

(b) TTL負荷

図9　バスタイミングテスト負荷

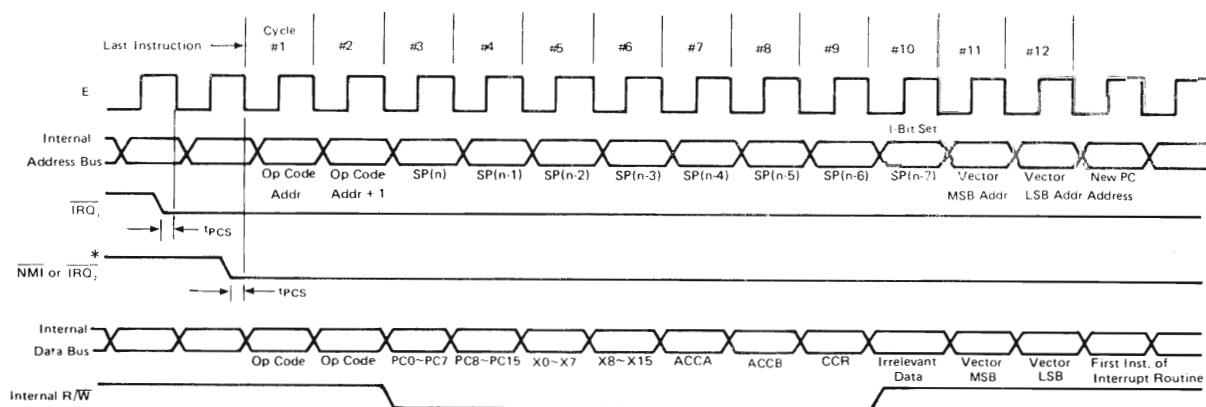



図10　インタラプトシーケンス

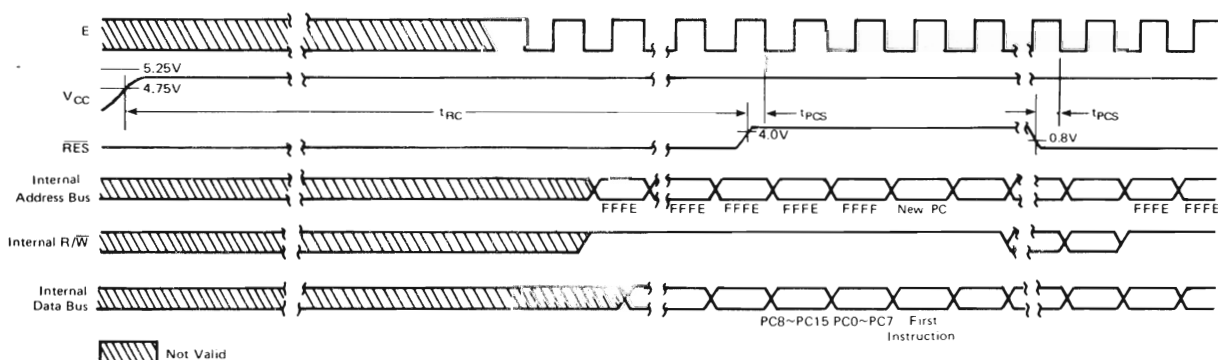



図11　リセットタイミング

■端子機能の説明

●V_{CC}, V_{SS}

この2本の端子は，電源とGNDに使用します。電源推奨電圧は，＋5V±5％です。

●XTAL, EXTAL

この端子にはATカットの並列共振形水晶振動子を接続します。4分周回路が内蔵されているのでシステムクロック1MHz を得るためには4 MHz の水晶振動子を使用します。時間的に厳しい用途でない場合には，TVの色回路で用いる3.58 MHz 水晶振動子を利用いただくと経済的です。安定な発振を得るためには，両端子とGND間にそれぞれ22 pF のコンデンサを挿入してください。

EXTAL端子は，外部から50％（±10％）の4 MHzクロックで駆動可能で，このときLSI内部では1 MHz のシステムクロックが得られます。駆動周波数は4 MHz 以下で使用可能です。外部駆動を使用する場合XTAL端子は接地してください。図12に接続推奨回路を示します。

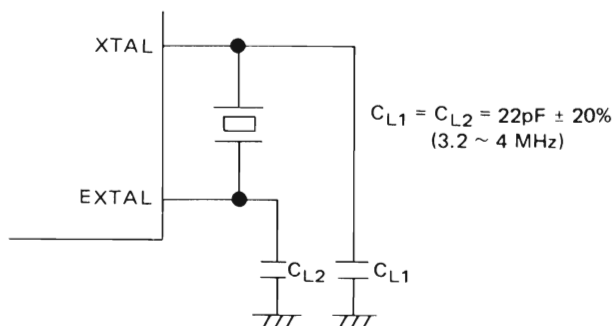



図12　水晶の接続推奨回路

●V_{CC} Standby

この端子には，電源V_{CC} ダウン時にスタンバイRAMを保持するため，電源電圧±5％を印加します。RAMの前半64バイトの内容は，これにより電源オフ時に保持されます（8mA max）。図13の回路は，電源オフ時にV_{CC}スタンバイ電圧を保持する回路の例です。電源オフ時にRAMの内容を保持するには，次の手続きが必要です。

1）RAMイネーブルビット(RAME)に“0”を書込んでください。RAMEは，$0014番地のRAMコントロールレジスタのビット6です。この操作がスタンバイRAM部分をディスエーブルしますので，電源オフに際して内容が保存されます。

2）V_{CC} スタンバイ電圧をV_{SBB}(min) 以上に保持してください。

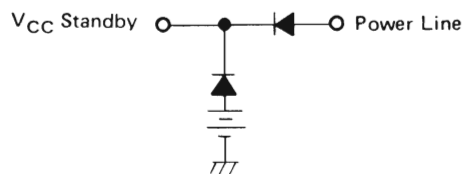



図13　V_{CC} スタンバイのための電池バックアップの例

●リセット（$\overline{RES}$）

この端子は電源オフ状態から，MCUをリセット，スタートさせるために使用します。電源投入時には$\overline{RES}$端子は，最低100 ms の間 “Low” レベルに保持してください。動作中にリセットする場合には，少なくとも3システムクロック周期分は “Low” レベルにしてください。この端子に “High” レベルが検出されると，MCUは次のように動作します。

1）アドレスラインの上位8ビットをすべて “High” レベルにします。

2）I/Oポート2のビット2, 1, 0の状態をプログラムコントロールレジスタのビットPC2，PC1，PC0にラッチします。

3）最後の2つの番地（$FFFE，$FFFF）の内容（スタートアドレス）がプログラムカウンタに入り，その番地からプログラムがスタートできるようにします（表1参照）。

4）割込みマスクビットがセットされます。MPUがマスク可能な割込み$\overline{IRQ_1}$を認識できるようにするためには，事前にクリアしてください。

●イネーブル（E）

内部発振器が使用されている場合，この端子から外部へシステムクロックを供給します。出力は，単相でTTLコンパチブルであり，水晶発振周波数の1/4のクロック周波数となります。この端子は1個のTTL負荷と90 pF の容量を駆動することができます。

●ノンマスカブルインタラプト（$\overline{NMI}$）

この端子の入力信号の立ち下がりが検出されると，MCU内部でノンマスカブル割込みシーケンスが開始されます。次に述べる$\overline{IRQ_1}$割込みの場合と同様に，$\overline{NMI}$信号検出時に実行中の命令は最後まで実行が続けられます。コンディションコードレジスタ中の割込みマスクビットは，$\overline{NMI}$にまったく影響を与えません。

$\overline{NMI}$割込みへの応答が開始されると，インデックスレジスタ，プログラムカウンタ，アキュムレータ，コンディションコードレジスタの内容は，スタック領域に退避されます。このシーケンスが終了すると，ベクタアドレス$FFFC，$FFFDを発生し，この2つの番地の内容をプログラムカウンタへ読込み，ノンマスカブル

割込みサービスルーチンへブランチします。

割込み端子のワイアードORをとり，最適な割込み制御を行う場合には，3.3kΩの外付け抵抗を V_{CC}と端子の間に挿入してください。

入力端子 $\overline{IRQ_1}$と$\overline{NMI}$は，イネーブル信号でサンプルされるハードウェア上の割込み端子で，命令実行終了後，E同期で割込みルーチンを発生させます。

●インタラプトリクエスト（$\overline{IRQ_1}$）

これは，レベル検出形端子であり，MPU内で割込みシーケンスを発生させる要求端子です。MPUは割込み要求が入ったとき実行中の命令があれば，それが終了するまで要求の受付けを待ちます。要求検知時に，コンディションコードレジスタ内の割込みマスクビットがセットされていなければ，MPUは割込みシーケンスを開始しますが，セットされていれば割込み要求は無視されます。シーケンス開始後，インデックスレジスタ，プログラムカウンタ，アキュムレータ，コンディションコードレジスタの内容がスタック領域へ退避されます。次に割込みマスクビットをセットし，これ以後はマスカブル割込みを受付けないようにします。サイクルの最後にメモリ番地＄FFF8と＄FFF9を指す16ビットベクタアドレスを発生し，この番地に入っている内容をプログラムカウンタへ読込み，MPUは割込みサービスルーチンへブランチします。

表1　割込ベクタ用メモリマップ

	Vector		Interrupt
	MSB	LSB	
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$ (or $\overline{IS3}$)
	FFF6	FFF7	ICF (Timer Input Capture)
	FFF4	FFF5	OCF (Timer Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

割込みをワイアードORして最適状態で使用するには，$\overline{IRQ_1}$ 端子と V_{CC} の間に3.3kΩを挿入してください。内部割込みは，内部割込み信号（$\overline{IRQ_2}$）を使用します。この割込みは，＄FFF0～＄FFF7のベクタアドレスを使う点を除いて$\overline{IRQ_1}$ とまったく同様です。

$\overline{IRQ_1}$ は$\overline{IRQ_2}$ に対して，同時割込み要求発生時に優先権をもちます。コンディションコードレジスタ中の割込みマスクビットは，セットされていると両方の割込みを禁止します。

以下の説明の端子は，シングルチップモードの場合のポート3にのみ適用されます。

●インプットストローブ（$\overline{IS3}$）（SC_1）

この信号の機能は，I/Oポート3コントロールステータスレジスタによって変ります。$\overline{IS3}$ イネーブルビットがセットされていれば，$\overline{IS3}$ 信号の立ち下がりによって割込みが起動されます。また，ラッチイネーブルビットがセットされていれば，I/Oポート3上のデータがポート3のデータレジスタにラッチされます。正常な入力データラッチのために必要な信号のタイミング条件は，図6に示すとおりです。

●アウトプットストローブ（$\overline{OS3}$）（SC_2）

この信号は，プロセッサが外部デバイスに対し，有効データがI/O端子上にあることを示すためのストローブ信号に使用されます。アウトプットストローブのタイミング特性は，図5に示します。

以下の説明の端子はエクスパンディッドモードの場合に適用されます。

●リード／ライト（R／$\overline{W}$）（SC_2）

このTTLコンパチブル出力信号は，ペリフェラルやメモリに対しMCUがリード（"High"）かライト（"Low"）かどちらの状態なのかを示します。この信号の通常スタンバイ状態は，リード（"High"）です。この出力は，1個のTTL負荷と90 pF のコンデンサを駆動できます。

●I/Oストローブ（$\overline{IOS}$）（SC_1）

エクスパンディッドノンマルチプレックスモードでの動作時には，$\overline{IOS}$によって，A_9～A_{15}は"0"として，A_8は"1"として内部的にデコードされます。この結果，＄0100から＄01FFまでの256番地の外部メモリのアクセスが可能になります。タイミング特性は，図2に示します。

●アドレスストローブ（AS）（SC_1）

エクスパンディッドマルチプレックスモードでの動作時に，アドレスストローブがこの端子に出力されます。この信号は，ポート3上にデータとマルチプレックスして出力されるアドレス下位8ビットをラッチするために使用され，図19に示すようにアドレスストローブを使って，8ビットのラッチを動かします。

これにより，Eパルス期間中は，I／Oポート3をデータバスとして利用することができます。

この信号のタイミング特性は，図1に示します。

■ポート

HD68P01 MCUには4つのI/Oポート（3つの8ビットポートと1つの5ビットポート）があります。8ビットポートの1つには，2本の制御端子が付いています。各々のポートは個別に書込み専用のデータディレクションレジスタを持ち，個々のI/O端子を入力用または出力用にプログラムできます*。

対応するデータディレクションレジスタのビットが“1”の場合はI/O端子は出力にプログラムされ，“0”の場合は入力にプログラムされます。4つのポートはそれぞれポート1，ポート2，ポート3，ポート4とします。4つのポートのアドレスと，その対応するデータディレクションレジスタのアドレスは表2に示すとおりです。

*唯一の例外はポート2のビット1で，この端子はデータ入力またはタイマ出力のどちらかになります。このビットは出力ポートとしては使用できません。

表2　ポートおよびデータディレクションレジスタのアドレス

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

●I／Oポート1

これは8ビットのポートで，各々のビットは対応するデータディレクションレジスタの内容により，入力用または出力用に定められます。

8ビットの出力バッファはスリーステートバッファになっており，この端子が入力として使用される場合には，高インピーダンス状態になります。正確なリードのためには，入力端子上の電圧はロジック“1”のときは2.0V以上，ロジック“0”のときは0.8V以下としなければなりません。

これらの端子は，TTLコンパチブルです。また，1.5Vで1mA以内の駆動源として使用でき，ダーリントントランジスタのベースを直接駆動することができます。MCUリセット後は，I／O端子はすべて入力となっています。

3種類のモードのいずれの場合も，ポート1は常に並列I/O端子として動作します。

●I／Oポート2

このポートは5端子から成っており，そのデータディレクションレジスタにより，入出力方向が定められます。

5ビットの出力バッファはスリーステートバッファになっており，入力として使用する場合には，高インピーダンス状態になります。正確なリードのためには，入力端子上の電圧はロジック“1”のときは2.0V以上，ロジック“0”のときは0.8V以下としなければなりません。

出力の場合，内部にプルアップ抵抗を有していませんが，TTLを直接駆動できます。CMOS入力を駆動する場合は，外部プルアップ抵抗が必要です。MCUリセット後は，I/O端子はすべて入力となっています。

ポート2の3本の端子（8,9,10ピン）は，リセット期間中，モードプログラミング用に使用されます。リセット時のこの3端子の値は，ポート2の上位3ビット（ビット5,6,7）にラッチされます。これについてはモード選択の項で説明します。

3種類のモードのいずれにおいてもポート2はI/O端子として働くことができ，シリアルコミュニケーションインタフェースやタイマに対するインタフェースにもなります。ただし，ビット1（P_{21}）は，データ入力またはタイマ出力に限定されます。

●I／Oポート3

これは8ビットより成るポートで，I/O端子，データバスまたはデータバスとマルチプレックスされたアドレスバスとして使用されます。その機能は，リセット期間中にユーザよりハードウェア的にポート2の3ビットを使用してプログラムされたモードに依存します。データバスとしてのポート3は双方向性です。

ペリフェラルデバイスから入力する場合は，標準TTLレベルでなければなりません。つまり，ロジック“1”のときには2.0V以上，ロジック“0”のときには0.8V以下とします。

このTTLコンパチブルスリーステートバッファは1個のTTL負荷と90 pFが駆動できます。

エクスパンディッドモードの場合は，リセット後データディレクションレジスタは禁止され，データの流れは，R/$\overline{\text{W}}$信号の状態に依存します。

3種類のモードにおいてポート3は次の機能を持っています。

シングルチップモード

パラレル入/出力となり，対応するデータディレクションレジスタによって，入力または出力として定義されます。

このモードに関連した2本の制御端子はインプットストローブ（$\overline{\text{IS3}}$）とアウトプットストローブ（$\overline{\text{OS3}}$）として，ハンドシェイクに使われます。この制御端子は，I/Oポートコントロール／ステータスレジスタによって制御されます。

ポート3の3種類の付加機能を次にまとめます。

1）ポート3の入力データは$\overline{\text{IS3}}$（SC_1）を制御信号として使用し，ラッチすることができます。
2）$\overline{\text{OS3}}$（SC_2）は，ポート3のデータレジスタの読出し，または書込み動作によって発生します。
3）$\overline{\text{IRQ}_1}$の割込みは$\overline{\text{IS3}}$の立ち下がりによって発生

させることができます。
ポート3のラッチおよびストローブタイミングは，図5，図6に示しています。
エクスパンディッドノンマルチプレックスモード
このモードのときには，ポート3はデータバス(D_0～D_7)になります。
エクスパンディッドマルチプレックスモード
この場合には，ポート3はデータバス(D_0～D_7)とアドレスバスの下位8ビット(A_0～A_7)がマルチプレックスされて使用されます。

I/Oポート3　コントロールステータスレジスタ

	7	6	5	4	3	2	1	0
$000F	$\overline{IS3}$ FLAG	$\overline{IS3}$ $\overline{IRQ_1}$ ENABLE	X	OSS	LATCH ENABLE	X	X	X

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 LATCH ENABLE.

このビットは，I/Oポート3の入力ラッチを制御します。このビットが"1"にセットされている場合，ポート3上の入力データは，インプットストローブ$\overline{IS3}$の立ち下がりでラッチされます。なお，このラッチはI/Oポート3のリード動作により，再びラッチ可能な状態になります。このラッチイネーブルビットは，リセットによってクリアされます。
Bit 4 OSS (アウトプットストローブ選択)
このビットは，アウトプットストローブがI/Oポート3へのライト動作により発生するのか，ポート3からのリード動作により発生するのかを選択します。このビットがクリアされているときは，ストローブはポート3のリードにより発生します。ビットがセットされているときは，ポート3へのライトによりストローブが発生します。
Bit 5 Not used.
Bit 6 $\overline{IS3}$ ENABLE.
このビットがセットされているとき，$\overline{IS3}$フラグ(ビット7)がセットされていれば割込みはイネーブルされ，クリアすると割込みは禁止されます。このビットは，$\overline{RES}$によりクリアされます。
Bit 7 $\overline{IS3}$ FLAG.
これはインプットストローブ$\overline{IS3}$(SC_1)の立ち下がりでセットされる読出し専用ビットです。コントロール/ステータスレジスタの読出しに続いて行われるI/Oポート3のリード/ライト動作により，このビットはクリアされます。このビットはリセットによりクリアされます。

● I/Oポート4
これは8ビットポートで，動作モードによりI/O端子またはアドレス出力端子になります。正確なリード動作のためには，入力端子上の電圧はロジック"1"のときには2.0V以上，ロジック"0"のときには0.8V以下としなければなりません。
出力の場合，各々の端子はTTLコンパチブルで1個のTTL負荷と90 pFを駆動できます。リセット後は，このポートは入力となります。したがってこの端子をアドレスとして使用するためには，ポートを出力としてプログラムする必要があります。
3種類のモードにおいてポート4は以下のようになります。
シングルチップモード
対応するデータディレクションレジスタによりプログラムされるパラレル入/出力となります。
エクスパンディッドノンマルチプレックスモード
このモードでは，ポート4はデータディレクションレジスタに"1"を書込むことにより，下位アドレス(A_0～A_7)出力端子になります。
アドレスとして必ずしも8ビット全部を必要としない場合は，残りの端子を上位ビットから順にI/O端子(入力のみ)として使用できます。
エクスパンディッドマルチプレックスモード
このモードでは，ポート4はデータディレクションレジスタに"1"を書込むことにより，上位アドレス(A_8～A_{15})出力端子になります。アドレスとして，必ずしも8ビット全部を必要としない場合は，残りの端子を上位ビットから順にI/O端子(入力のみ)として使用できます。

■モードの選択
HD68P01のリセット後の動作するモードは，ユーザが端子10,9,8をハードウェア的に外部で配線することにより決定しなければなりません。この3端子はポート2の下位ビットI/O2，I/O1，I/O0です。
これら3つの信号はリセットが"High"になるときに，I/Oポート2レジスタのプログラムコントロールビットPC2，PC1，PC0にラッチされます。
I/Oポート2レジスタは以下に示すとおりです。

ポート2　データレジスタ

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

モード選択に使用する外部ハードウェアの例を図14に示します。HD14053Bはリセット期間中，ペリフェラルデバイスとMCUの間を分離するために利用できます。これはペリフェラルデバイスとモード発生回路間でデータ

が競合する場合必要となります。

ポート2のビット5,6,7は，リード専用なので，モードはソフトウェアでは変えられません。表3にHD68P01のモード選択をまとめます。

HD68P01は3つの基本モードで動作します。

1）シングルチップモード
2）エクスパンディッドマルチプレックスモード
（HMCS6800周辺LSIとコンパチブル）
3）エクスパンディッドノンマルチプレックスモード

●シングルチップモード

シングルチップモードでは，ポートすべてがI／O機能になります。図16にシングルチップモードを示します。

このモードでは，ポート3のSC_1，SC_2端子は制御端子として機能し，データをハンドシェイクするためのインプットストローブ信号入力$\overline{IS3}$と，アウトプットストローブ信号出力$\overline{OS3}$として利用できます。

●エクスパンディッドノンマルチプレックスモード

このモードではHD68P01は外付ロジックなしで，直接HMCS6800ペリフェラルにアドレスします。またポート3はデータバスに，ポート4はA_0～A_7アドレスバスまたはアドレスバスの一部とI／O（入力のみ）として機能します。ポート2はパラレルI/O，またはシリアルI/O，またはタイマおよびこれらの組合わせ機能として働きます。ポート1はパラレルI/Oとしてのみ使用できます。このモードでは，HD68P01は最大256番地までアドレスすることができます。アドレスが少なくてよい応用システムでは，ポート4の残りの端子をI/O端子（入力のみ）として使用できます（図17参照）。

●エクスパンディッドマルチプレックスモード

このモードでは，ポート4はI／O端子（入力のみ），またはアドレス端子として機能します。ポート3にはデータバスと下位アドレスバスがマルチプレックスされ，これはアドレスストローブ出力により分離できます。ポート2は，5本のパラレルI/O端子またはSCI，またはタイマおよびこれらの組合わせ機能として働きます。ポート1は8本のパラレルI／O端子として機能し，最大65kワードのアドレス空間まで拡張できます。

●下位アドレスバスラッチ

エクスパンディッドマルチプレックスモードのときには，データバスはポート3に下位アドレスバスとマルチプレックスされるので，アドレスビットを外部でラッチしなければなりません。ラッチには8個のD形フリップフロップを備えた74LS373を用いて，下位アドレスバイトをラッチすることができます。図19にHD68P01と74LS373の接続法を示します。

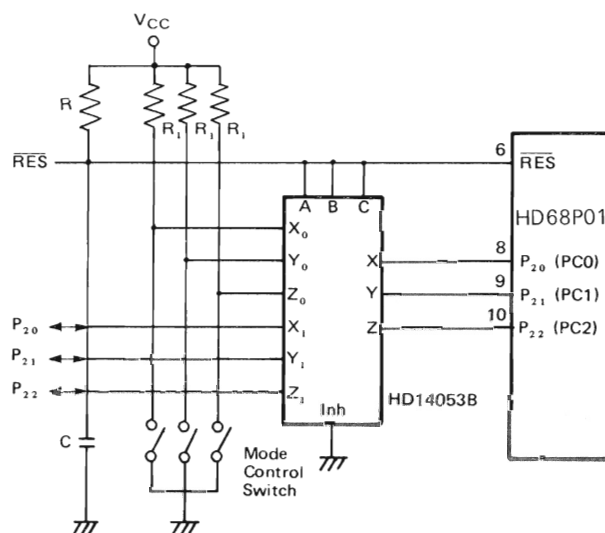
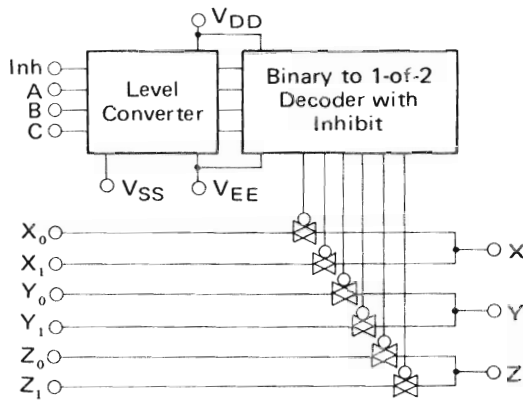



図14　モード選択のための推奨回路

真理値表

Control Input				On Switch		
Inhibit	Select					
	C	B	A	HD14053B		
0	0	0	0	Z_0	Y_0	X_0
0	0	0	1	Z_0	Y_0	X_1
0	0	1	0	Z_0	Y_1	X_0
0	0	1	1	Z_0	Y_1	X_1
0	1	0	0	Z_1	Y_0	X_0
0	1	0	1	Z_1	Y_0	X_1
0	1	1	0	Z_1	Y_1	X_0
0	1	1	1	Z_1	Y_1	X_1
1	X	X	X	–		

図15　HD14053Bマルチプレクサ／デマルチプレクサ

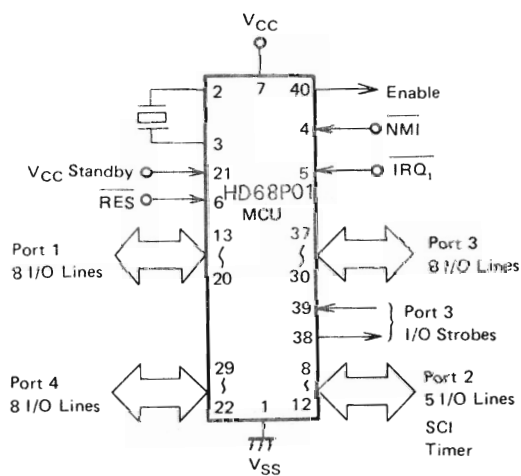



図16　HD68P01 MCUシングルチップモード

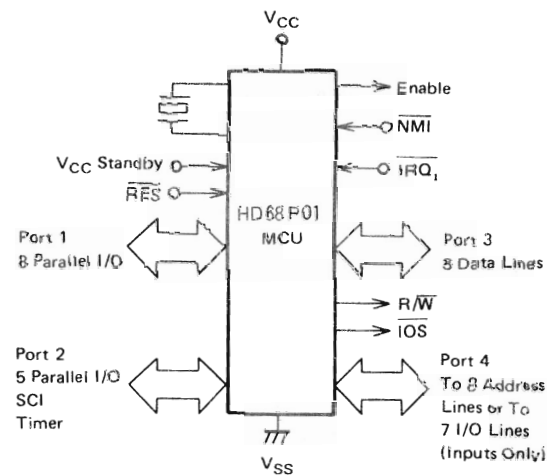



図17　HD68P01 MCUエクスパンディッドノンマルチプレックスモード

図18　HD68P01 MCUエクスパンディッドマルチプレックスモード

真理値表

Output Control	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

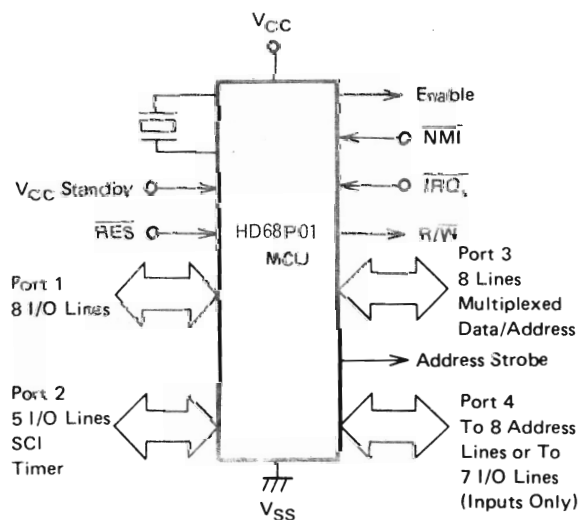
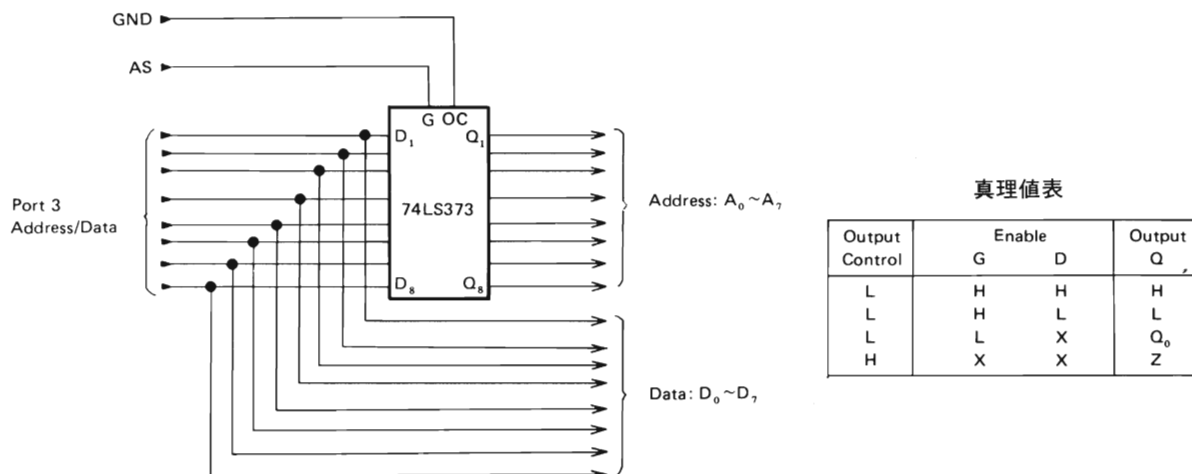



図19　ラッチ接続

●モードとポートの概要およびMCU信号

ここで各モードにおけるMCUの信号について説明します。SC_1とSC_2はチップのモード変化に伴い変化する信号です。

MODE	PORT 1 Eight Lines	PORT 2 Five Lines	PORT 3 Eight Lines	PORT 4 Eight Lines	SC_1	SC_2
SINGLE CHIP	I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)
EXPANDED MUX	I/O	I/O	ADDRESS BUS (A_0~A_7) DATA BUS (D_0~D_7)	ADDRESS BUS* (A_8~A_{15})	AS(O)	R/$\overline{W}$(O)
EXPANDED NON-MUX	I/O	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS* (A_0~A_7)	$\overline{IOS}$(O)	R/$\overline{W}$(O)

＊これらの端子は，アドレスラインとして不要の場合上位から順に，I/O端子(入力専用)として使用することができます。

I=Input　　$\overline{IS3}$=Input Strobe　　SC=Strobe Control
O=Output　　$\overline{OS3}$=Output Strobe　　AS=Address Strobe
R/$\overline{W}$=Read/Write　　$\overline{IOS}$=I/O Select

表3　モードの選択

Mode	P_{22} (PC2)	P_{21} (PC1)	P_{20} (PC0)	ROM	RAM	Interrupt Voctors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX	Multiplexed/No RAM & ROM
2	L	H	L	E	I	E	MUX	Multiplexed/RAM
1	L	L	H	I	I	E	MUX	Multiplexed/RAM & ROM
0	L	L	L	I	I	I(3)	MUX	Multiplexed Test

凡例：
I — Internal
E — External
MUX — Multiplexed
NMUX — Non-Multiplexed
L — Logic "0"
H — Logic "1"

(注)
1) 内部RAMは$0080でアドレスされます。
2) 内部ROMは使用できなくなります。
3) $\overline{RES}$ベクタは$\overline{RES}$が"High"になった後，2サイクル間，外部になります。
4) ポート3，4に関係するアドレスはモード0，1，2，3では外部とされます。
5) ポート3に関係するアドレスはモード5，6では外部とされます。
6) ポート4のアドレス出力のうち，不要の部分は，入力指定して，入力ポートとして使用することができます。

■メモリマップ

HD68P01の内蔵ROM空間は8kバイトであり，動作モードによって最大65kバイトまでアドレスすることができます。また，EPROMの適用により，マップの使用空間が異なります。すなわち，

1）HD68P01S0

2kバイトROMのため，INTERNAL ROMの先頭アドレスは$F800としなければなりません（$E000，$E800，$F000，$F800が重なっているため)。

2）HD68P01V05, HD68P01V07

4kバイトROMのため，INTERNAL ROMの先頭アドレスは$F000としなければなりません（$E000，$F000が重なっているため)。

3）HD68P01M0

HN482764を使用すれば8kバイトROM内蔵となりこの場合のINTERNAL ROMの先頭アドレスは$E000からとなります。

各動作モードにおけるメモリマップを図20に示します。マップ先頭の32番地は，表4に示すように，内部レジスタ専用になっています。

表4　内部レジスタ

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* モード0,1,2,3,5,6では外部アドレスとなり，さらにモード5ではアクセスできません。
** モード0,1,2,3では外部アドレスとなります。
*** 1=Output, 0=Input

（注）
1）外部で使用できるアドレス$04, $05, $06, $07, $0Fは含みません。
2）アドレス$FFFE, $FFFFは$\overline{RES}$の立ち上がりエッジ後,2サイクル以内でアクセスした場合は外部, その他の場合は内部になります。
3）2MPUサイクル後, データバスが1デバイス以上ドライブするのを避けるために, 内部および外部のメモリ空間のオーバラップがあってはいけません。
4）このモードは, 外部リセットベクタを使用して内部ROMの割込みベクタのテストの場合のみに使用します。

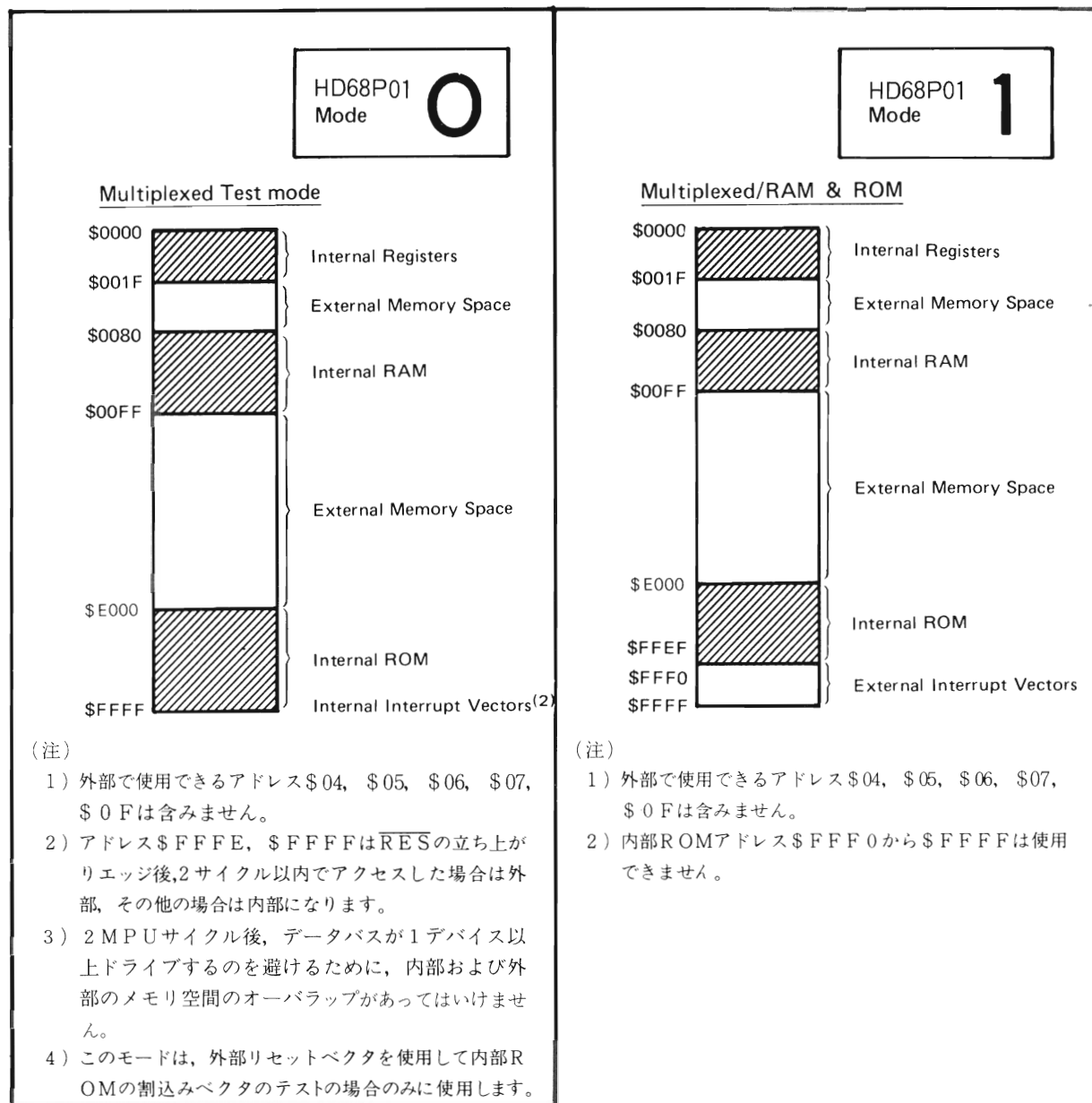
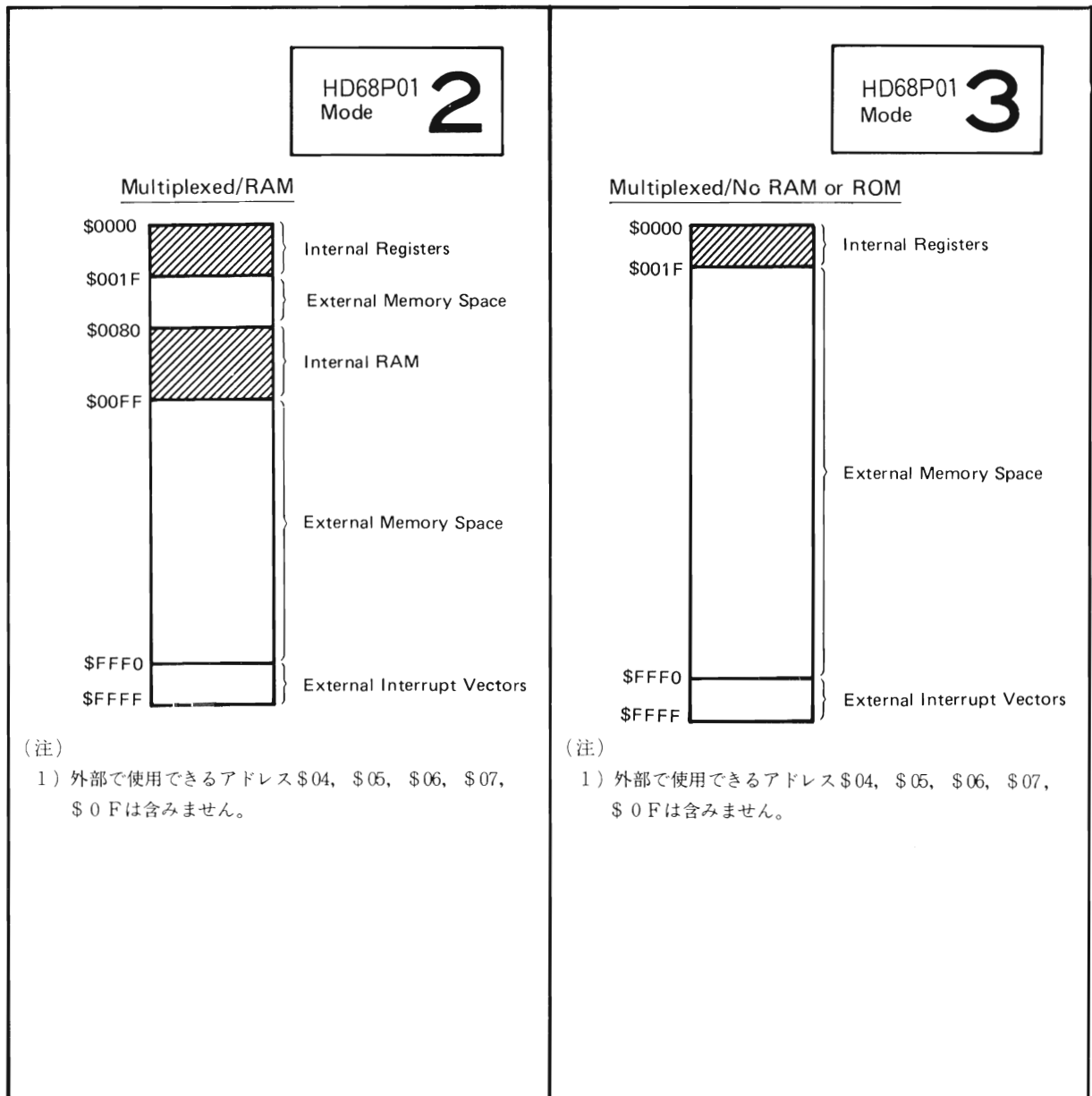
（注）
1）外部で使用できるアドレス$04, $05, $06, $07, $0Fは含みません。
2）内部ROMアドレス$FFF0から$FFFFは使用できません。

（次頁に続く）

図20　HD68P01 メモリマップ

（次頁に続く）

図20　HD68P01メモリマップ

（注）

1）内部ＲＯＭは使用できなくなります。

2）モード４は，ポート２のデータレジスタのＰＣ０ビットに“1”を書込むことによってリセットを挿入せずに，モード５に変わります。

3）アドレスA_8～A_{15}は“don't cares”として扱われ，内部ＲＡＭにデコードされます。

4）内部ＲＡＭは，$××80～＄××ＦＦにあらわれます。

（注）

1）＄04，＄06，＄０Ｆは含みません。なお，これらのレジスタは外部でも使用できません。

2）このモードへは，(1)$\overline{\text{RES}}$によるモードプログラム(2)$\overline{\text{RES}}$によりモード４に設定した後で，ポート２データレジスタのＰＣ０ビットに“1”を書込むの２つの方法により入ることができます。

3）アドレス端子A_0～A_7は，ポート４に対してのデータディレクションレジスタの適応するビットに“1”が書込まれるまで，アドレスは含みません。これらのアドレス端子はデータディレクションレジスタを書込むことによって出力されるまで，“High”レベルとなっています。

（次頁に続く）

図20　ＨＤ68P01メモリマップ

（注）

1）外部で使用できるアドレス＄04，＄06，＄０Ｆは含みません。

2）アドレス端子A_8～A_{15}は，ポート４のデータディレクションレジスタの適応するビットに“１”が書込まれるまで，アドレスを含みません。これらのアドレス端子は，データディレクションレジスタを書込むことによって出力されるまで，“High”レベルとなっています。

図20　ＨＤ68Ｐ01メモリマップ

■プログラマブルタイマ

HD68P01は16ビットのプログラマブルタイマを内蔵しており，入力波形の測定に使用できます。またそれとは別に，独立に出力波形を発生することができます。

入力，出力の両波形共，パルス幅は数マイクロ秒から数秒の間で変化させることができます。

タイマハードウェアは次のものから構成されます。

・8ビットのコントロール／ステータスレジスタ
・16ビットのフリーランニングカウンタ
・16ビットのアウトプットコンペアレジスタ
・16ビットのインプットキャプチャレジスタ

図21にタイマのブロックダイアグラムを示します。

●フリーランニングカウンタ（$0009：000A）

プログラマブルタイマの主要部分は，16ビットのフリーランニングカウンタで，これはE（イネーブル）クロックより駆動され，その値をインクリメントします。カウンタ値は，いつでもMPUソフトウェアで読出しができます。

カウンタはリセットでクリアされます。またこのカウンタは，リード専用レジスタと考えてさしつかえありません。

ただしMPUからライトが行われると，ライトデータの値のいかんにかかわらず，カウンタ（$09，$0A）にはプリセット値$FFF8が書込まれます。この機能はMCUのテストのためのものですが，応用することもできます。

●アウトプットコンペアレジスタ（$000B：000C）

アウトプットコンペアレジスタは，16ビットのリード/ライトレジスタで出力波形制御に使用されます。このレジスタの内容はフリーランニングカウンタの値と常時比較されています。

内容が一致すると，タイマコントロール/ステータスレジスタ（TCSR）の中のフラグ（OCF）がセットされ，TCSR中のアウトプットレベルビット（OLVL）のそのときの値がポート2のビット1に送られます。ポート2用データディレクションレジスタのビット1が“1”（出力）となっている場合には，この値がポート2のビット1に出力されます。その後，次の比較値でアウトプットレベルを再び制御するために，アウトプットコンペアレジスタの値とアウトプットレベルビットの値を変更します。なお，アウトプットコンペアレジスタはリセットすると$FFFFにセットされます。アウトプットコンペアレジスタの上位バイトへの書込みサイクル直後の1サイクル間は，比較機能が禁止されます。これはレジスタに16ビットの値を確実にセットしてから比較を開始するためです。*

*コンペアレジスタへのデータ書込みはSTX命令などの2バイト転送命令を使用してください。

●インプットキャプチャレジスタ（$000D：000E）

インプットキャプチャレジスタは，16ビットのリード専用レジスタで，外部入力信号が変化したときのフリーランニングカウンタの値を保持します。

これをトリガするのに必要な入力の変化の極性は，TCSRのインプットエッジビット（IEDG）により制御されます。

外部入力信号がエッジ検出回路に入力されるためには，ポート2のビット0に対応するデータディレクションレジスタのビットがクリア（“0”）されていなければなりません。*

すべての場合にインプットキャプチャを保障するには，入力パルス幅は少なくとも2イネーブルサイクルは必要です。

*ポート2のビット0を出力とし，“1”にセットしても，外部入力信号はエッジ検出回路によって検知されます。

●タイマコントロール／ステータスレジスタ（TCSR）（$0008）

タイマコントロール／ステータスレジスタは8ビットのレジスタです。8ビットすべてリード可能で，下位5ビットはライトも可能です。上位3ビットはリード専用で，以下のタイマステータス情報を示します。

1）入力が指定した極性の変化をして，カウンタ値がインプットキャプチャレジスタに転送されました（ICF）。
2）カウンタ値とアウトプットコンペアレジスタの値が一致しました（OCF）。
3）カウンタ値が$0000になっています（TOF）。

この各フラグはTCSR中に個々のイネーブルビットをもち，これにより，内部割込み信号（$\overline{IRQ_2}$）に割込み要求を出力するか否かがコントロールされます。コンディションコードレジスタ中のIビットがクリアされていれば，これらのフラグがセットされるのに応じて，優先ベクタアドレスを発生します。各ビットの説明は次のとおりです。

タイマコントロール／ステータスレジスタ

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

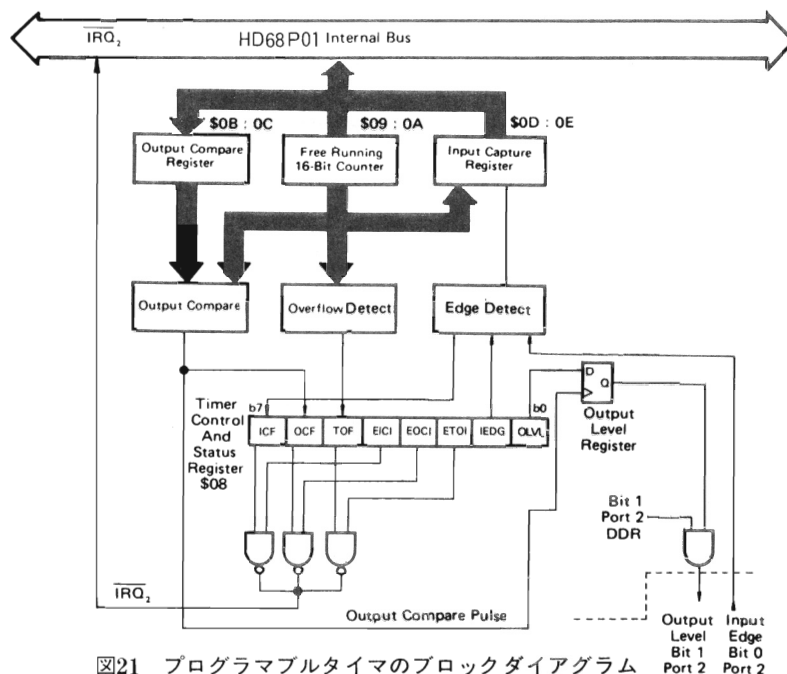



図21　プログラマブルタイマのブロックダイアグラム

Bit 0 OLVL アウトプットレベル

この値は，カウンタ値とアウトプットコンペアレジスタ値の比較が一致したとき，ポート2のビット1に移されます。

ポート2のビット1に対応するＤＤＲが“1”にセットされていれば，この値はポート2のビット1端子に出力されます。

Bit 1　IEDG　インプットエッジ

このビットは，カウンタからインプットキャプチャレジスタへのデータ転送を，ポート2のビット0入力端子の変化のどちらの極性でトリガするかを決定します。この機能を使うときは，ポート2のビット0に対応するＤＤＲを事前にクリアしておく必要があります。

IEDG＝0のとき入力の立ち下がりエッジでトリガされます(“High” から “Low” への変化)。

IEDG＝1のとき入力の立ち上がりエッジでトリガされます(“Low” から “High” への変化)。

Bit 2　ETOI　イネーブルタイマオーバフローインタラプト

このビットがセットされている場合は，ＴＯＦ割込みにより内部割込み($\overline{IRQ_2}$)が発生するのを許しますが，クリアされている場合は，割込み発生が禁止されます。

Bit 3　EOCI　イネーブルアウトプットコンペアインタラプト

このビットがセットされている場合には，ＯＣＦ割込みにより内部割込み($\overline{IRQ_2}$)が発生するのを許可し,クリアされている場合には割込みの発生を禁止します。

Bit 4　EICI　イネーブルインプットキャプチャインタラプト

このビットがセットされている場合，ＩＣＦ割込みにより，内部割込み($\overline{IRQ_2}$)が発生するのを許可し，クリアされている場合，割込み発生を禁止します。

Bit 5　TOF　タイマオーバフローフラグ

このリード専用ビットは,カウンタ値が$FFFFになったときセットされ，ＴＣＳＲをリードした後，引続きカウンタ($0009)をリードしたときクリアされます。

Bit 6　OCF　アウトプットコンペアフラグ

このリード専用ビットは，アウトプットコンペアレジスタ値とカウンタ値が一致したときセットされ，ＴＣＳＲのリードに続いて，アウトプットコンペアレジスタ($000B or $000C)へＭＣＵがライトしたときクリアされます。

Bit 7　ICF　インプットキャプチャフラグ

このリード専用ビットは，エッジ検出回路への入力信号がＩＥＤＧで指定した変化をした場合にセットされ，ＴＣＳＲのリードに続いて，インプットキャプチャレジスタ($000D)をＭＣＵがリードしたときクリアされます。

■シリアルコミュニケーションインタフェース

HD68P01はフルデュプレックスのアシンクロナスシリアルコミュニケーションインタフェース（SCI）を内蔵しています。SCIは，数種類のデータ速度選択が可能です。送信部と受信部から構成されており，各々独立に動作可能ですが，双方とも同一データフォーマットと同一データ転送速度で動作します。送／受信部双方共データバスを介してMPU部とデータのやりとりを行い，外部とは，ポート2のビット2,3,4を介して通信します。

ハードウェア，ソフトウェア，レジスタについて以下に説明します。

●ウェイクアップ機能

一般的なマルチプロセッサ応用の場合，ソフトウェアプロトコールは，通常行先番地をメッセージの先頭バイトに持っています。非選択MPUにメッセージの残りの部分を無視させるためにウェイクアップ機能があります。これによって非選択MPUは，それ以後のすべての割込み処理を次のメッセージ開始まで禁止することができます。

このハードウェア機能により非選択MPUは次のメッセージが来たとき，そのメッセージに対して再びイネーブル状態にされます（ウェイクアップされます)。

ウェイクアップ機能は，10ビットの連続した“1”の送信信号によってトリガされます。

したがってソフトウェアプロトコールは，メッセージ間において，アイドル期間を与えるようにしておく必要があります。

●プログラマブルオプション

HD68P01のシリアルI/O部の以下の機能はプログラマブルになっています。

- データフォーマット……標準マーク／スペース（NRZ）
- クロック源……外部または内部
- ボーレート……MCUのϕ_2に対して4種のレートの中の1つまたは，外部クロックの1/8
- ウェイクアップ機能……イネーブルまたはディスエーブル
- 割込み要求……トランスミットデータレジスタ，レシーブデータレジスタそれぞれに対し独立にイネーブルまたはマスクします。
- クロック出力……内部クロックのポート2のビット2への出力をイネーブルまたはディスエーブルします。
- ポート2（ビット3,4)……送信部と受信部用にポート2のビット3,4を使用するか否か。

●シリアルコミュニケーション部のハードウェア

シリアルコミュニケーション部のハードウェアは，図22に示すように4本のレジスタによって制御されます。

それは次のとおりです

- 8ビットのコントロール／ステータスレジスタ
- 4ビットの転送レート／モードコントロールレジスタ（ライト専用）
- 8ビットのリード専用レシーブデータレジスタ
- 8ビットのライト専用トランスミットデータレジスタ

これら4本のレジスタの他にシリアルI/O部は，ポート2のビット3（入力）とビット4（出力）を使用します。ポート2のビット2は，内部クロックを外部へ出力するオプションまたは，外部クロックを内部へ入力するオプションを選んだときに利用されます。

●トランスミット／レシーブコントロールステータスレジスタ（TRCSR）

TRCSレジスタは8ビットより成り，8ビット全部がリードできます。また0～4ビットはライトも可能です。このレジスタは$\overline{RES}$信号で，$20にイニシャライズされます。TRCSレジスタのビットは以下のように定められています。

トランスミット／レシーブコントロールステータスレジスタ

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR:$0011

Bit 0 WU ウェイクアップ

このビットはソフトウェアでセットされ，10個の連続した“1”データを受けたときハード的にクリアされます。ただし，このビットをセットする場合，REフラグはすでにセットされていなければなりません。

Bit 1 TE トランスミットイネーブル

このビットがセットされると，9個の連続した“1”データで構成されるプリアンブルを発生させ，引き続きポート2のビット4に送信部データを出力できるようになります。これは，このビットに対応するDDRのビットの値に無関係に行われます。

TEビットがクリアされているときには，シリアルI/Oは，ポート2のビット4に何ら影響を与えません。ただし，TEビットのセットは，転送レート／モードコントロールレジスタのセット後，少なくとも転送レートの1ビット期間後に行ってください。

Bit 2 TIE トランスミットインタラプトイネーブル

このビットがセットされている場合，TDRE（ビット5）がセットされたときに内部割込み（$\overline{IRQ_2}$）が発生するのを許可します。クリアされている場合，TDREの内部割込み（$\overline{IRQ_2}$）発生はマスクされます。

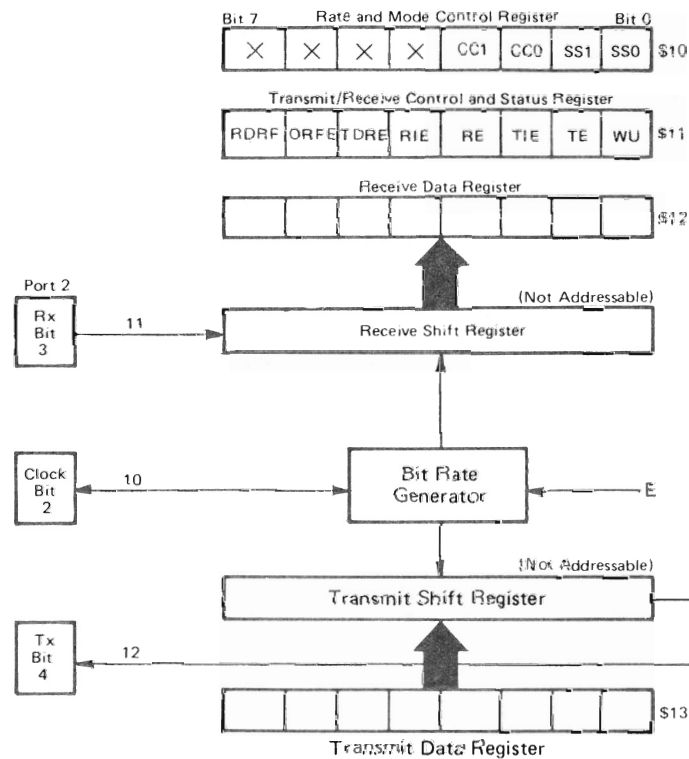



図22 シリアルI／Oレジスタ

Bit 3 RE レシーブイネーブル

このビットがセットされている場合，このビットに対応するＤＤＲの内容の如何にかかわらずポート２のビット３信号を受信部に入力します。クリアされていればシリアルＩ／Ｏは，ポート２のビット３に何ら影響を与えません。

Bit 4 RIE レシーブインタラプトイネーブル

このビットがセットされているときは，ビット７（ＲＤＲＦ）またはビット６（ＯＲＦＥ）がセットされると内部割込み（$\overline{\mathrm{IRQ}_2}$）が発生するのを許可します。クリアされているときには，割込み発生はマスクされます。

Bit 5 TDRE トランスミットデータレジスタエンプティ

トランスミットデータレジスタからアウトプットシフトレジスタへデータ転送が行われたとき，ハードウェア的にセットされます。このビットは，ステータスレジスタのリード後，トランスミットデータレジスタに次の新しいバイトデータを書込むとリセットされます。

ＴＤＲＥは，$\overline{\mathrm{RES}}$により“1”に初期値設定されます。

Bit 6 ORFE オーバランフレーミングエラー

このビットはオーバランやフレーミングエラーが発生したとき（受信時のみ）ハード的にセットされます。

オーバランエラーはRDRFがセットされたままの状態で新しいバイトデータを，レシーブデータレジスタに転送しようとしたとき発生します。フレーミングエラーは，ビットカウンタとビットストリームのバイトの境目が同期しなかった場合に発生します。ただし，ウェイクアップフラグがセットされている場合（WU＝１）には，このビットはセットされません。このビットは，ステータスレジスタリード後，レシーブデータレジスタをリードするとクリアされます。また$\overline{\mathrm{RES}}$によってもクリアされます。

Bit 7 RDRF レシーブデータレジスタフル

このビットは，レシーブシフトレジスタからレシーブデータレジスタへのデータ転送が行われると，ハード的にセットされます。ただし，ウェイクアップフラグがセットされている場合（WU＝１）には，このビットはセットされません。このビットはステータスレジスタを読出した後でレシーブデータレジスタを読出すとクリアされます。また，$\overline{\mathrm{RES}}$によってもクリアされます。

転送レート/モードコントロールレジスタ

7	6	5	4	3	2	1	0
X	X	X	X	CC1	CC0	SS1	SS0

ADDR : $0010

表5　SCIビット時間および転送レート

		XTAL	2.4576 MHz	4.0 MHz
SS1	SS0	E	614.4 kHz	1.0 MHz
0	0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud
0	1	E ÷ 128	208μs/4,800 Baud	128 μs/7812.5 Baud
1	0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud
1	1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud

表6　SCIフォーマットおよびクロックソースコントロール

CC1, CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
00	—	—	—	—	—
01	NRZ	Internal	Not Used	**	**
10	NRZ	Internal	Output*	**	**
11	NRZ	External	Input	**	**

* TRCSレジスタのビットREとTEの内容の如何にかかわらずクロック出力されます。
** TRCSのREビットが"1"ならばビット3はシリアル入力として使用されます。
TRCSのTEビットが"1"ならばビット4はシリアル出力として使用されます。

●転送レート／モードコントロールレジスタ(RMCR)

このレジスタは次のシリアルI／O部の変数をコントロールします。

・ボーレート　　・データフォーマット
・クロック源　　・ポート2のビット2の機能

このレジスタは4ビットライト専用レジスタで，$\overline{RES}$によりクリアされます。この4ビットは，2ビットづつの2フィールド構成と考えることができます。すなわち下位の2ビットは，内部クロックのビットレートをコントロールし，上位2ビットはデータフォーマットとクロック源選択ロジックをコントロールします。

Bit0 SS0 ｝スピードセレクト
Bit1 SS1

これらのビットは，内部クロックのときのボーレートを選択します。選択可能な4つのレートはMPUのϕ_2クロック周波数の関数となります。表5に利用できるボーレートを示します。

Bit2 CC0 ｝クロックコントロール／フォーマット選択
Bit3 CC1

これらのビットは，データフォーマットとクロック源選択ロジックをコントロールします。

表6にビットの定義を示します。

●内部発生クロック

ユーザがシリアルI／O用に内部クロックを外部で使用する場合には次の点に留意してください。

・RE，TEの値は関係しません。
・CC1，CC0は"10"にセットしなければなりません。
・利用できる最大クロックレートはE÷16となります。
・クロックはビットレートの1倍で，ビットの中間で立ち上がります。

●外部発生クロック

もしユーザがシリアルI／O用に外部クロックを供給する場合には，次の点に留意してください。

・RMCレジスタ中のCC1，CC0は"11"にセットしなければなりません。
・外部クロックは使用するボーレートの8倍にセットされなければなりません。
・最大外部クロック周波数は1.0MHzです。

●シリアル動作

シリアルI／Oハードウェアは，動作させるのに先立

ち，HD68P01のソフトウェアにより，イニシャライズしなければなりません。このシーケンスは通常以下のようになっています。

・RMCレジスタ中に希望する動作コントロールビットを書込みます。
・TRCSレジスタに希望する動作コントロールビットを書込みます。

　ポート2のビット3，4をシリアルI／O専用に使用する場合はTE，REビットはセットしたままの状態にしておいて結構です。

● 送信動作

データ送信は，TRCSレジスタ中のTEビットのセットにより可能になります。このビットがセットされますと，シリアルトランスミットシフトレジスタの内容をポート2のビット4に出力します。ポート2ビット4は対応するDDR値に依存せず無条件に出力になります。

$\overline{\text{RES}}$に続いてユーザは，RMCレジスタとTRCSレジスタの両方を希望動作条件にセットする必要があります。この手続き中にTEビットをセットすると10ビットの“1”よりなるプリアンブルを送信します。プリアンブル出力後内部同期が確立され，送信部分は動作可能状態となります。

この時点において次の2つの状態のどちらかが行われます。

1）もしも送信データレジスタが“空”（TDRE＝1）ならば“1”の連続値が送出され，アイドル状態であることを示します。
2）もしデータがトランスミットデータレジスタに入っていれば（TDRE＝0），データはアウトプットシフトレジスタに送られデータの送信が開始されます。

データ送信期間では，先ず“0”のスタートビットが送出されます。次に8ビットデータ（ビット0から始まります。）とストップビットが送出されます。

トランスミットデータレジスタが空になると，ハードウェアがTDREフラグビットをセットします。

もし，MCUが適正時間内にフラグに応答できなかった場合（TDREはパラレルデータレジスタからシリアルアウトプットレジスタへの次の正常なデータ転送が行われるときまではセットされたままになっています。），スタートビットの“0”の代わりに“1”が送出され，さらにデータレジスタにデータが供給されるまでは引き続き“1”状態が続けて出力されます。TDREが“1”になっている間は，“0”が送出されることはありません。

● 受信動作

受信動作は，REビットのセットにより可能になります。これによりポート2のビット3はシリアル入力になります。受信部分の動作条件はTRCSレジスタとRMCレジスタの内容により決められます。レシーバのビット間隔は，内部同期のために8つの副間隔に分けられます。通常の非バイフェイズモードでは，受信ビットの流れは，最初に入ってくる“0”（スペース）により同期化されます。10ビットの期間は各ビットともほぼその真中でストローブされます。もし第10番目のビットが1（ストップビット）でなければ，フレーミングエラーと判定されORFEビットがセットされます。

もし第10番目のビットが1ならば，データはレシーブデータレジスタに転送され，割込みフラグRDRFがセットされます。もし次のデータの第10番目のビットが受信されてもなおRDRFがまだセットされたままならば，ORFEがセットされ，オーバランが発生したことを示します。

MCUがRDRFフラグまたはORFEフラグに対する応答としてステータスレジスタの読出しの後，データレジスタの読出しを行うと，RDRFまたはORFEはクリアされます。

■ RAMコントロールレジスタ

＄0014番地に割付けられたこのレジスタは，スタンバイRAMの状態についての情報を与えます。RAMイネーブルビット（RAME）が“0”であると，スタンバイRAMをディスエーブルし，V_{CC}スタンバイの信号の項で説明したように，V_{CC} StandbyがV_{SBB}電圧より高く保持されていれば，RAM内容を電源オフ時に保持します。

RAM　コントロールレジスタ

	7	6	5	4	3	2	1	0
$0014	STBY PWR	RAME	X	X	X	X	X	X

Bit 0　Not used.
Bit 1　Not used.
Bit 2　Not used.
Bit 3　Not used.
Bit 4　Not used.
Bit 5　Not used.
Bit 6　**RAME RAM** イネーブル

ユーザは，このコントロールビットを使いスタンバイRAMをディスエーブルすることができます。MPUをリセットするとこのビットには，“1”がセットされ，スタンバイRAMをイネーブルにします。このビットは，プログラム制御により“1”または“0”を書込むことができます。

RAMがディスエーブルの状態（ロジック“0”）では，RAMアドレスは無効となり，MPUは，外部メモリからデータをリードすることができます。

Bit 7 STBY PWR スタンバイパワービット

このビットはリード／ライト両用のビットで，スタンバイRAMのデータが有効であるかどうかを判断するフラグです。このビットがセットされていればスタンバイRAMのデータは有効です。V_{CC} Standby端子に印加されている電圧がV_{SBB}(min)以下になり，スタンバイRAMのデータが有効でなくなるとクリアされます。このビットはソフトウェアでのみセットされ，リセットには影響されません。

■命令セット概要

HD68P01はHMCS6800の全命令セットを活用できるようにするため，HD6800のアップワードコンパチブルのオブジェクトコードを持っており，また主要命令の実行時間はスループットの向上を図るため改善されています。

さらに16ビットデータ操作命令とハードウェア乗算命令を含む新しい命令も付加されています。

この項では，以下のものについて説明します。

- MCUプログラミングモデル（図23参照）
- アドレッシングモード
- アキュムレータとメモリの操作命令（表7参照）
- 新命令系
- インデックスレジスタとスタックの操作命令（表8参照）
- ジャンプおよびブランチ命令（表9参照）
- コンディションコードレジスタ操作命令（表10参照）
- 命令実行マシンサイクル数（表11参照）
- 命令のサイクルごとの動作概要（表12参照）
- 未定義命令の動作概要
- オペコードマップ（表13参照）

●MCUプログラミングモデル

HD68P01のプログラミングモデルを図23に示します。ダブルアキュムレータ(D)は，構造的にはアキュムレータBとアキュムレータAを結合したものなので，アキュムレータDを使う命令では，A，Bの内容はこわされます。

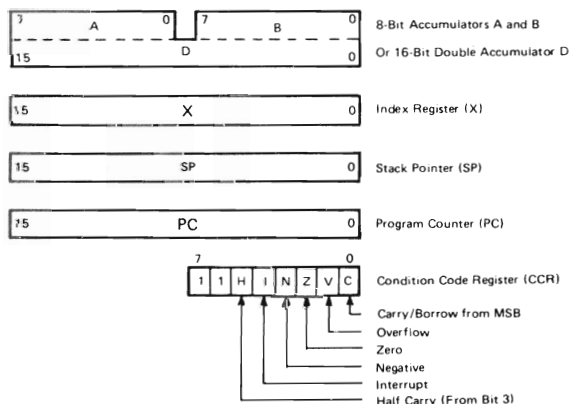



図23 MCUプログラミングモデル

●MCUアドレッシングモード

HD68P01は，7つのアドレッシングモードを持っています。アドレッシングモードは，命令タイプとコードの両方で決められます。命令ごとのアドレッシングモードについては，表11の中にマシンサイクル数で示した命令実行時間と共に示します。

クロック周波数が4 MHzのときは，マシンサイクルがそのままマイクロ秒になります。

アキュムレータ（ACCX）アドレッシング

アキュムレータのみがアドレッシングされ，アキュムレータAまたはBのいずれかが指定されます。これらは1バイト命令です。

イミディエイトアドレッシング

このアドレッシングではオペランドが命令の第2バイトに含まれますが，LDSとLDXは例外で第2，第3のバイトにオペランドが含まれます。実行の際イミディエイト命令をフェッチするとき，MPUはこの番地をアドレスします。これは，2または3バイト命令です。

ダイレクトアドレッシング

このアドレッシングでは，命令の第2バイトがオペランドを格納している番地を示します。これにより0～255番地の256バイト領域を直接アドレスすることができます。これらの領域にデータを格納すると実行時間の短縮が図られますので,システムを構成するときはこの領域をRAM構成とし，ユーザのデータ領域として使用することをお奨めします。これは2バイト命令です。

エクステンディッドアドレッシング

この命令では第2バイトがオペランドの格納されているアドレスの上位8ビットを示し，第3バイトが下位8ビットを示します。この番地はメモリ中の絶対番地を示します。これは，3バイト命令です。

インデックスドアドレッシング

このアドレッシングでは，命令の第2バイトの内容と，インデックスレジスタの下位8ビットが加算されます。さらにこのキャリは，インデックスレジスタの上位8ビットに加えられます。その結果がメモリをアドレスするために使用されます。この修飾されたアドレスは，テンポラリアドレスレジスタに保持されますので，インデックスレジスタの内容は変化しません。これは2バイト命令です。

インプライドアドレッシング

このアドレスモードでは，命令そのものがアドレスを与えます。すなわち,スタックポインタ，インデックスレジスタなどをアドレスします。これは1バイト命令です。

リラティブアドレッシング

このアドレッシングでは，命令の第2バイトの内容と，プログラムカウンタの下位8ビットが加算されます。そのキャリまたはボローが上位8ビットに加算されます。そのためユーザは，現在実行中の命令の－126～＋129バイトの範囲内でデータをアドレスすることができます。これは2バイト命令です。

表7　アキュムレータ，メモリ操作命令

Operations	Mnemonic	IMMED			DIRECT			INDEX			EXTEND			IMPLIED			Boolean/ Arithmetic Operation	5 H	4 I	3 N	2 Z	1 V	0 C
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#							
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 − M → M	•	•	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	10	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Msp, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	3	1	B → Msp, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3				M	•	•	↕	↕	⑥	↕
	ROLA													49	2	1	A — C b7 b0	•	•	↕	↕	⑥	↕
	ROLB													59	2	1	B	•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				M	•	•	↕	↕	⑥	↕
	RORA													46	2	1	A — C b7 b0	•	•	↕	↕	⑥	↕
	RORB													56	2	1	B	•	•	↕	↕	⑥	↕

Condition Code Register

（注）コンディションコードレジスタの説明は表10で後述します。

（次頁に続く）

表7　アキュムレータ，メモリ操作命令

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M, A, B: C ← b7 … b0 ← 0	●	●	↕	↕	⑥	↕
	ASLA													48	2	1		●	●	↕	↕	⑥	↕
	ASLB													58	2	1		●	●	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	3	1	C ← ACC A/ ACC B (A7 A0 B7 B0) ← 0	●	●	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M, A, B: b7 … b0 → C	●	●	↕	↕	⑥	↕
	ASRA													47	2	1		●	●	↕	↕	⑥	↕
	ASRB													57	2	1		●	●	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M, A, B: 0 → b7 … b0 → C	●	●	R	↕	⑥	↕
	LSRA													44	2	1		●	●	R	↕	⑥	↕
	LSRB													54	2	1		●	●	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	3	1	0 → ACC A/ ACC B (A7 A0 B7 B0) → C	●	●	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A − M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B − M → B	●	●	↕	↕	↕	↕
Double Subtract	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				A : B − M : M + 1 → A : B	●	●	↕	↕	↕	↕
Subtract Accumulators	SBA													10	2	1	A − B → A	●	●	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A − M − C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B − M − C → B	●	●	↕	↕	↕	↕
Transfer Accumulators	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test Zero or Minus	TST							6D	6	2	7D	6	3				M − 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A − 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B − 00	●	●	↕	↕	R	R

（注）コンディションコードレジスタの説明は表10で後述します。

●追加命令

従来のHMCS6800命令セットに加えて，HD68P01は，次の新命令を持っています。

ABX……IX⇐IX+ACCB

アキュムレータBの8ビットの符号なしデータを16ビットのインデックスレジスタにキャリを考慮して加算し，インデックスレジスタに結果を格納します。

ADDD……ACCD⇐(ACCD)＋(M：M+1)

倍精度アキュムレータD*にメモリ番地M：M+1の倍精度データを加算し，結果をACCDに格納します。

* ACCDは，アキュムレータA，Bを結合させて作られた16ビットレジスタ（A：B）です。アキュムレータAは最上位バイトです。

ASLD……ACCDすべてのビットを左へ1ビットシフトします。

ビット0には“0”が入ります。Cビット（キャリ）にはACCDの最上位ビットにあった内容が入ります。

LDD……ACCD⇐(M：M+1)

M：M+1に格納されている倍精度データを倍精度アキュムレータDに格納します。

コンディションコードはデータ内容によって決まります。

LSRD……ACCDの全ビットを右へ1ビットシフトします。

ビット15には，“0”が入ります。Cビット（キャリ）には，ACCDの最下位ビットの内容が入ります。

MUL……ACCD⇐ACCA×ACCB

ACCAとACCBの8ビットデータを乗算し，符号なし16ビットの結果をA：Bに格納します。

ACCAには，結果の上位バイトを格納します。

PSHX……↓(IXL)，SP←(SP)−1
↓(IXH)，SP←(SP)−1

インデックスレジスタの内容は，スタックポインタの示すアドレスのスタックに退避され，スタックポインタは−2されます。

PULX……SP← (SP) +1：↑ (IXH)
　　　　　SP← (SP) +1：↑ (IXL)

スタックポインタ+1番地のスタックからインデックスレジスタに引き出され，スタックポインタは+2されます。

STD……M：M+1← (ACCD)

ACCDの内容をメモリに格納します。

ACCDの内容は変化しません。

SUBD……ACCD⇐ (ACCD) − (M：M+1)

ACCDの内容から，メモリ中の倍長データを減算し，結果をACCDに格納します。

BRN……ブランチしません。この命令は実行に3サイクル要する2バイトNOP命令と等価です。

表8　インデックスレジスタ，スタック操作命令

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M : M + 1	●	●	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X − 1 → X	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	3	1	SP − 1 → SP	●	●	●	●	●	●
Increment Index Reg	INX													08	3	1	X + 1 → X	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	3	1	SP + 1 → SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	●	●	⑦	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	●	●	⑦	↕	R	●
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	●	●	⑦	↕	R	●
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M+1)	●	●	⑦	↕	R	●
Index Reg → Stack Pntr	TXS													35	3	1	X − 1 → SP	●	●	●	●	●	●
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	●	●	●	●	●	●
Add	ABX													3A	3	1	B + X → X	●	●	●	●	●	●
Push Data	PSHX													3C	4	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	●	●	●	●	●	●
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	●	●	●	●	●	●

（注）コンディションコードレジスタについての詳細は表10に後述します。

表9　ジャンプ/ブランチ命令

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	●	●	●	●	●	●
Branch Never	BRN	21	3	2													None	●	●	●	●	●	●
Branch If Carry Clear	BCC	24	3	2													C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS	25	3	2													C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ	27	3	2													Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE	2C	3	2													$N \oplus V = 0$	●	●	●	●	●	●
Branch If > Zero	BGT	2E	3	2													$Z + (N \oplus V) = 0$	●	●	●	●	●	●
Branch If Higher	BHI	22	3	2													C + Z = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE	2F	3	2													$Z + (N \oplus V) = 1$	●	●	●	●	●	●
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT	2D	3	2													$N \oplus V = 1$	●	●	●	●	●	●
Branch If Minus	BMI	2B	3	2													N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC	28	3	2													V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS	29	3	2													V = 1	●	●	●	●	●	●
Branch If Plus	BPL	2A	3	2													N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR	8D	6	2														●	●	●	●	●	●
Jump	JMP							6E	3	2	7E	3	3				See Special Operations	●	●	●	●	●	●
Jump To Subroutine	JSR				9D	5	2	AD	6	2	BD	6	3					●	●	●	●	●	●
No Operation	NOP													01	2	1	Advances Prog. Cntr. Only	●	●	●	●	●	●
Return From Interrupt	RTI													3B	10	1					⑧		
Return From Subroutine	RTS													39	5	1	See Special Operations	●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait for Interrupt	WAI													3E	9	1		●	⑨	●	●	●	●

表10　コンディションコードレジスタ操作命令

Operations	Mnemonic	AddressingModes			Boolean Operation	Condition Code Register					
		IMPLIED				5	4	3	2	1	0
		OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	●	●	●	●	●	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	●	R	●	●	●	●
Clear Overflow	CLV	0A	2	1	0 → V	●	●	●	●	R	●
Set Carry	SEC	0D	2	1	1 → C	●	●	●	●	●	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	●	S	●	●	●	●
Set Overflow	SEV	0B	2	1	1 → V	●	●	●	●	S	●
Accumulator A → CCR	TAP	06	2	1	A → CCR				⑩		
CCR → Accumulator A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

（注）コンディションコードレジスタ（テスト結果が真であるときセットされ，そうでなければクリアされます。）

① (Bit V)　テスト：結果＝10000000？
② (Bit C)　テスト：結果≠00000000？
③ (Bit C)　テスト：上位バイトのＢＣＤキャラクタが10以上か？（事前にセットされていればクリアされません。）
④ (Bit V)　テスト：実行前のオペランド＝10000000？
⑤ (Bit V)　テスト：実行前のオペランド＝01111111？
⑥ (Bit V)　テスト：命令実行後　$N \oplus C = 1$ でセット
⑦ (Bit N)　テスト：結果が０より小さいか？（ビット15＝１）
⑧ (All)　スタックからコンディションコードレジスタへロードします。
⑨ (Bit I)　割込み発生時にセット，あらかじめセットされている場合この待ち状態から抜けるには，ノンマスカブル割込みが必要です。
⑩ (全ビット)　アキュムレータＡの内容にしたがってセットされます。
⑪ (Bit C)　乗算の結果ＡＣＣＢのビット７＝１か？

表11　命令実行マシンサイクル数

	ACCX	Imme-diate	Direct	Ex-tended	In-dexed	Im-plied	Re-lative
ABA	●	●	●	●	●	2	●
ABX	●	●	●	●	●	3	●
ADC	●	2	3	4	4	●	●
ADD	●	2	3	4	4	●	●
ADDD	●	4	5	6	6	●	●
AND	●	2	3	4	4	●	●
ASL	2	●	●	6	6	●	●
ASLD	●	●	●	●	●	3	●
ASR	2	●	●	6	6	●	●
BCC	●	●	●	●	●	●	3
BCS	●	●	●	●	●	●	3
BEQ	●	●	●	●	●	●	3
BGE	●	●	●	●	●	●	3
BGT	●	●	●	●	●	●	3
BHI	●	●	●	●	●	●	3
BIT	●	2	3	4	4	●	●
BLE	●	●	●	●	●	●	3
BLS	●	●	●	●	●	●	3
BLT	●	●	●	●	●	●	3
BMI	●	●	●	●	●	●	3
BNE	●	●	●	●	●	●	3
BPL	●	●	●	●	●	●	3
BRA	●	●	●	●	●	●	3
BRN	●	●	●	●	●	●	3
BSR	●	●	●	●	●	●	6
BVC	●	●	●	●	●	●	3
BVS	●	●	●	●	●	●	3
CBA	●	●	●	●	●	2	●
CLC	●	●	●	●	●	2	●
CLI	●	●	●	●	●	2	●
CLR	2	●	●	6	6	●	●
CLV	●	●	●	●	●	2	●
CMP	●	2	3	4	4	●	●
COM	2	●	●	6	6	●	●
CPX	●	4	5	6	6	●	●
DAA	●	●	●	●	●	2	●
DEC	2	●	●	6	6	●	●
DES	●	●	●	●	●	3	●
DEX	●	●	●	●	●	3	●
EOR	●	2	3	4	4	●	●
INC	2	●	●	6	6	●	●
INS	●	●	●	●	●	3	●

	ACCX	Imme-diate	Direct	Ex-tended	In-dexed	Im-plied	Re-lative
INX	●	●	●	●	●	3	●
JMP	●	●	●	3	3	●	●
JSR	●	●	5	6	6	●	●
LDA	●	2	3	4	4	●	●
LDD	●	3	4	5	5	●	●
LDS	●	3	4	5	5	●	●
LDX	●	3	4	5	5	●	●
LSR	2	●	●	6	6	●	●
LSRD	●	●	●	●	●	3	●
MUL	●	●	●	●	●	10	●
NEG	2	●	●	6	6	●	●
NOP	●	●	●	●	●	2	●
ORA	●	2	3	4	4	●	●
PSH	3	●	●	●	●	●	●
PSHX	●	●	●	●	●	4	●
PUL	4	●	●	●	●	●	●
PULX	●	●	●	●	●	5	●
ROL	2	●	●	6	6	●	●
ROR	2	●	●	6	6	●	●
RTI	●	●	●	●	●	10	●
RTS	●	●	●	●	●	5	●
SBA	●	●	●	●	●	2	●
SBC	●	2	3	4	4	●	●
SEC	●	●	●	●	●	2	●
SEI	●	●	●	●	●	2	●
SEV	●	●	●	●	●	2	●
STA	●	●	3	4	4	●	●
STD	●	●	4	5	5	●	●
STS	●	●	4	5	5	●	●
STX	●	●	4	5	5	●	●
SUB	●	2	3	4	4	●	●
SUBD	●	4	5	6	6	●	●
SWI	●	●	●	●	●	12	●
TAB	●	●	●	●	●	2	●
TAP	●	●	●	●	●	2	●
TBA	●	●	●	●	●	2	●
TPA	●	●	●	●	●	2	●
TST	2	●	●	6	6	●	●
TSX	●	●	●	●	●	3	●
TXS	●	●	●	●	●	3	●
WAI	●	●	●	●	●	9	●

●各サイクルの動作概要

表12は，命令ごと，サイクルごとのアドレスバス，データバス，リード／ライト信号（R／$\overline{W}$）の動きの詳細をまとめたものです。この情報は，制御プログラムが正常に実行されているかどうかをデバックするときに，ソフト，ハード両面から期待値と実測データを比較するのに役立ちます。アドレッシングモードと命令当たりのサイクル数により，分類してあります（一般には，同一アドレッシングモードで同一サイクル数の命令は同じ要領で実行されます。例外は表の中に示します）。

表12　命令の各サイクルごとの動作

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Operand Data
LDS LDX LDD	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 2 Address Bus FFFF	1 1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	Op Code Address of Operand Operand Data
STA	3	1 2 3	Op Code Address Op Code Address + 1 Destination Address	1 1 0	Op Code Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Op Code Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Subroutine Address Stack Pointer Stack Pointer + 1	1 1 1 0 0	Op Code Irrelevant Data First Subroutine Op Code Return Address (Low Order Byte) Return Address (High Order Byte)

（次頁に続く）

表12　命令の各サイクルごとの動作

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

*TST 命令のみ，6 サイクル目は次のようになります。AB=FFFF，R/$\overline{W}$=1，DB=Low Byte of Restart Vector.

（次頁に続く）

表12　命令の各サイクルごとの動作

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Address of Operand (High Order Byte)

*TST命令のみ，6サイクル目は次のようになります。AB=FFFF，R/$\overline{W}$=1，DB=Low Byte of Restart Vector.

（次頁に続く）

表12　命令の各サイクルごとの動作

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMPLIED					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address + 1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI*	9	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Op Code Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

* WAI命令によりWait中は次のようになります。AB=Stack Pointer−7, R/$\overline{W}$=1, DB=Irrelevant Data.

（次頁に続く）

表12　命令の各サイクルごとの動作

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
WAI		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMT BVS BRN	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

●未定義命令の動作概要

HD68P01は36種類の未定義命令があります。これらの未定義命令が実行されると，MPU内部のレジスタやメモリの内容などがランダムに変更されることがあります。また，4E，5Eのオペコードを実行するとプログラムカウンタの値がカウントアップされ続け，リセット信号が入力されるまでMPUは停止しません。このコードは，LSIのテスト時に使用するものです。

表13　オペコードマップ

HD68P01 MICROCOMPUTER INSTRUCTIONS																		
OP CODE						ACC A	ACC B	IND	EXT	ACCA or SP				ACCB or X				
										IMM	DIR	IND	EXT	IMM	DIR	IND	EXT	
HI		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								1
0010	2			BHI	PULA(+1)					SBC								2
0011	3			BLS	PULB(+1)	COM				*	SUBD (+2)			*	ADDD (+2)			3
0100	4	LSRD(+1)		BCC	DES	LSR				AND								4
0101	5	ASLD(+1)		BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX(+1)		BVC	PULX(+2)	ASL				EOR								8
1001	9	DEX(+1)	DAA	BVS	RTS(+2)	ROL				ADC								9
1010	A	CLV		BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI(+7)					ADD								B
1100	C	CLC		BGE	PSHX(+1)	INC				*	CPX (+2)			*	LDD (+1)			C
1101	D	SEC		BLT	MUL(+7)	TST				BSR (+4)	JSR (+2)			* (+1)	STD (+1)			D
1110	E	CLI		BGT	WAI(+6)	**		JMP(−3)		*	LDS (+1)			*	LDX (+1)			E
1111	F	SEI		BLE	SWI(+9)	CLR				* (+1)	STS (+1)			* (+1)	STX (+1)			F
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4	

(注) 1. ▱ は，未定義のオペコードです。
2. (　) 内の数字は，その命令のサイクルカウントに加えてください。
3. 下記の命令はすべて3バイト命令であり，表中に"*"で示しています。
SUBD, CPX, LDS, ADDD, LDD, LDX のイミディエイトアドレッシングモードおよび8F, CD, CF の未定義オペコード
4. 4E, 5E のオペコードは1バイト/∞サイクル命令であり，表中に"**"で示しています。

図24 割込みフローチャート

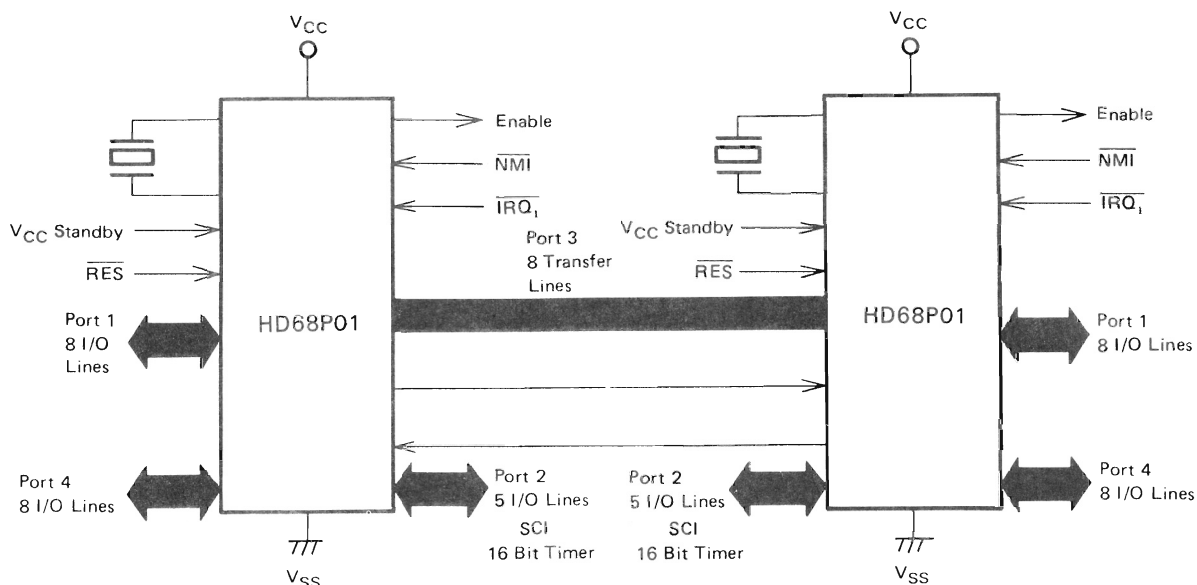



図25　HD68P01　MCUシングルチップデュアルプロセッサ構成

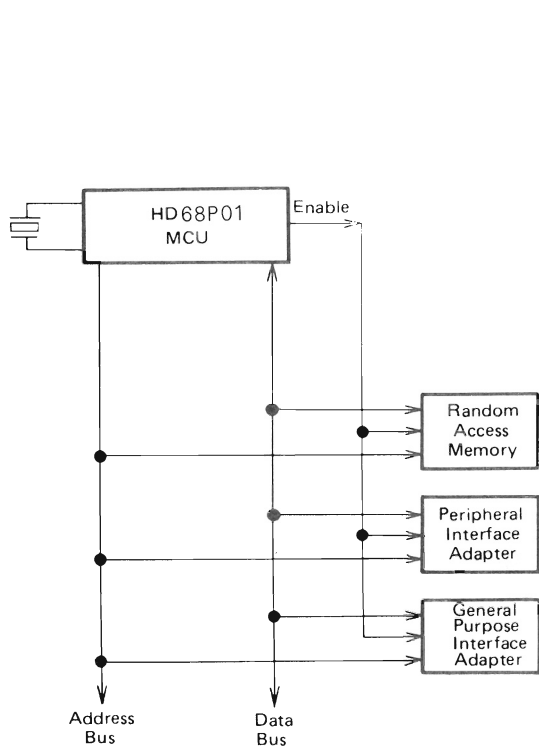



図26　HD68P01　MCUエクスパンディッドノンマルチプレックスモード

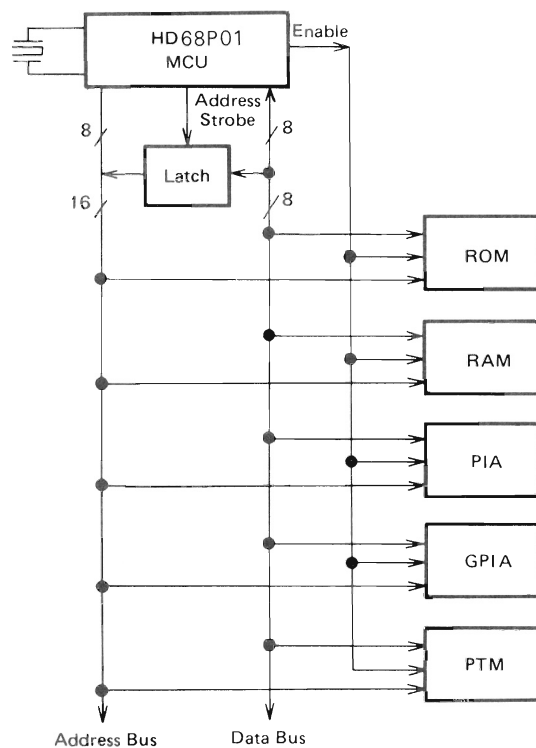



図27　HD68P01　MCUエクスパンディッドマルチプレックスモード

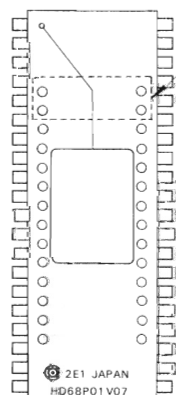
■HD6801S, HD6801V として使用する場合

HD68P01をHD6801Sとして使用するには，ROM使用空間をメモリアドレスの$F800番地から2kバイトまでとしてください。このときメモリアドレスの$E000番地から$F7FF番地までは使用できません。その他のメモリアドレス空間（65kワード以内）は，HD6801Sと同じであり，使用可能です。

HD68P01をHD6801Vとして使用するには，ROM使用空間をメモリアドレスの$F000番地から4kバイトまでとしてください。このときメモリアドレスの$E000番地から$EFFF番地までは使用できません。その他のメモリアドレス空間（65kワード以内）は，HD6801Vと同じであり，使用可能です。

■EPROM搭載形8ビットシングルチップマイクロコンピュータ使用上の注意

本製品はパッケージ上面にピンソケットを設けた特殊な構造をしていますので，使用する場合，次の点に留意してください。

(1)ピンソケット端子にはLSI端子と同様に静電気電圧，サージ電圧など最大定格以上の過電圧を印加しないように注意してください。LSIの永久破壊となることがあります。

(2)ピンソケットは28ピンありますが，16kおよび32kのEPROM（24ピン）を使用する場合はマーク捺印側につめて差込んでください。この場合，インデックス側の4ピンは開放状態にします。

(3)本製品をマスクROM形8ビットシングルチップマイクロコンピュータと同様にシステム製品に組込む場合には，EPROM端子とピンソケットとの接触性を良好に保つために次のことを守ってください。

(a) プリント基板などにはんだ付けする場合は，LSI端子の温度を250℃以下,時間は10秒以内で行うようにしてください。それ以上の温度あるいは長い時間熱しますとピンソケットを固定しているはんだが溶解し，ソケットが抜けることがあります。

(b) はんだ付後にフラックス洗浄を行うときや基板にコーティングを行うときに，洗浄液やコート剤がピンソケットに入らないよう注意してください。フラックスやコート剤によってピンソケットの接触性が劣化することがあります。

(c) 常時振動している場所，システムでの恒久的な使用はなるべく避けてください。

(d) EPROMの抜差しを繰り返したピンソケットは新しいものより接触性が損なわれています。システム製品に組込む場合にはなるべく新しい製品を使用することをおすすめします。

なお，不明の点は当社営業担当にご相談ください。

株式會社 日立製作所

電子事業本部
電子部品営業本部 〒100 東京都千代田区大手町二丁目6番2号（日本ビル） 電話 東京（03）270 2111（大代）

栃木電子部品営業所 西那須野（02873）6-3312
茨城電子部品営業所 勝田（0292）74-4011
多摩電子部品営業所 立川（0425）27-0410
関西支店 大阪（06）203-5781（代）
東北支店 仙台（0222）23-0121（代）
九州支店 福岡（092）741-5831（代）
金沢営業所 金沢（0762）63-2351（代）
中部支店 名古屋（052）251-3111（大代）
中国支店 広島（082）223-4111（代）
北海道支店 札幌（011）261-3131（大代）
四国支店 高松（0878）31-2111（代）

マイコンについての総合情報センター
日立マイコンセンター“GAIN”（03）253-1405（代）